

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Návrh a implementace rezervačního systému pro zimní stadion
Design and Implementation of Ice Rink Reservation System

Student: Bc. Aleš Kalousek
Vedoucí diplomové práce: Ing. Martin Pochyla, Ph.D.

Ostrava 2014

Zadání diplomové práce

Student: **Bc. Aleš Kalousek**

Studijní program: N6209 Systémové inženýrství a informatika

Studijní obor: 1802T001 Aplikovaná informatika

Téma: **Návrh a implementace rezervačního systému pro zimní stadion**
Design and Implementation of Ice Rink Reservation System

Zásady pro vypracování:

1. Úvod
2. Teoretické přístupy a nástroje tvorby informačních systémů a internetových prezentací
3. Analýza současného stavu a identifikace požadavků zadavatele
4. Návrh řešení a implementace rezervačního systému
5. Zhodnocení navrhovaného řešení
6. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků diplomové práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

KOFLER, Michael a Öggl BERND. *PHP 5 a MySQL 5*. Přeložil David ČEPIČKA. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.

NIXON, Robin. *Learning PHP, MySQL, JavaScript, and CSS*. 2nd Edition. Sebastopol: O'Reilly Media, 2012. ISBN 978-1-4493-1926-7.

SCHAFER, Steven M. *HTML, XHTML a CSS: Bible pro tvorbu WWW stránek*. Přeložil Marcel GOLIAŠ.

4. vyd. Praha: Grada Publishing, 2009. ISBN 978-80-247-2850-6.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Martin Pochyla, Ph.D.**

Datum zadání: 22.11.2013

Datum odevzdání: 25.04.2014

Ing. Petr Rozehnal, Ph.D.
vedoucí katedry



prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

„Prohlašuji, že jsem celou práci včetně všech příloh vypracoval samostatně.“

V Ostravě dne 23. 4. 2014


.....
Bc. Aleš Kalousek

Obsah

1. Úvod	5
2. Teoretické přístupy a nástroje tvorby informačních systémů a internetových prezentací	7
2.1. HTML a XHTML	7
2.1.1. HTML 5	9
2.2. CSS	12
2.2.1. CSS 3	14
2.3. JavaScript	15
2.3.1. ECMAScript	17
2.3.2. Objektový model dokumentu (DOM)	17
2.3.3. Objektový model prohlížeče (BOM)	18
2.3.4. JavaScript + HTML	18
2.4. jQuery	20
2.5. PHP	21
2.6. MySQL	25
2.7. Projektování informačních systémů	27
2.7.1. Identifikace a výběr projektů	28
2.7.2. Zahájení a plánování projektů	29
2.7.3. Analýza současného stavu	30
2.7.4. Návrh nového řešení	31
2.7.5. Zavedení nového řešení	32
2.7.6. Údržba systému	34
3. Analýza současného stavu a identifikace požadavků zadavatele	35
3.1. Popis objektu zadavatele	35
3.2. Analýza současného stavu	36

3.2.1. Požadavky zadavatele	37
4. Návrh řešení a implementace rezervačního systému	44
4.1. Návrh řešení	44
4.2. Implementace řešení	51
4.2.1. Integrace termínů rezervací do kalendářů různých platforem	60
5. Zhodnocení navrhovaného řešení	64
6. Závěr	67
Seznam použité literatury	70
Seznam zkratk	72
Prohlášení o využití výsledků diplomové práce	74
Seznam příloh	75

1. Úvod

Internet dnes patří mezi nedílné součásti každodenního života a jeho služby denně využívají masy lidí po celém světě. Jedná se o poměrně nové médium, uvážíme-li, že pro civilní účely funguje teprve od 80. let minulého století. Technologicky jde o počítačovou síť, lépe řečeno o systém navzájem propojených počítačových sítí. Smyslem existence takové sítě je poskytovat množství služeb jejím uživatelům. Mezi nejznámější služby v rámci internetu patří například elektronická pošta, online komunikace mezi uživateli, internetové telefonování, nebo přenos souborů. Tou nejpoužívanější službou je však bezpochyby WWW (World Wide Web). Jedná se o systém webových stránek, neboli o množinu navzájem propojených hypertextových dokumentů. Obrovské oblibě se tato služba těší především díky své uživatelské přívětivosti. Tato vlastnost je doložena tím, že uživatel bez problému zvládá obsluhou této složité technologie, aniž by znal podrobnosti o jejím fungování.

WWW je neustále se rozvíjející systém, který od svých počátků narostl do obrovských rozměrů. Původní koncept byl vytvořen na začátku 90. let 20. století organizací CERN (Evropská organizace pro jaderný výzkum) pro řešení specifických problémů s distribucí obrovského množství dat vědcům po celém světě. Tou dobou byl internet jen místem se statisíci propojenými počítači. Tim Berners-Lee (zaměstnanec CERN) vynalezl způsob komunikace mezi nimi pomocí výměny hypertextových dokumentů. Tento systém je znám jako Hyper Text Transfer Protocol (HTTP). Poté vytvořil také jazyk pro hypertext (HTML), první webový server a první webový prohlížeč. Dnes se zdají být tyto nástroje samozřejmostí, ale tehdy byl tento koncept revoluční a byly tak položeny základy celému systému World Wide Web.[NIXON,2012]

Jedním ze stavebních kamenů systému WWW je tedy také Hyper Text Transfer Protocol. Pomocí tohoto protokolu je definována komunikace mezi klientem a serverem. Vzhledem k tomu, že celý systém WWW je postaven na architektuře klient/server, je tato komunikace klíčová. Základním principem fungování HTTP protokolu je dotaz a odpověď. Klient (nejčastěji internetový prohlížeč) zašle zprávu webovému serveru v textové formě, server požadavek zpracuje a výsledek vrátí zpět klientovi, opět textovou formou. HTTP protokol je tzv. bezstavový, což znamená, že jednotlivé dotazy spolu nijak nesouvisí.

Další klíčovou technologií systému WWW jsou samotné webové stránky. V podstatě se jedná o textové dokumenty, které jsou definovány pomocí jazyku HTML (XHTML). Samotný text těchto dokumentů je doplněn o speciální značky a odkazy na jiné dokumenty. Primárně lze webové stránky rozdělit na tzv. statické a tzv. dynamické. Statické webové stránky, jak již název napovídá, jsou neměnné a zobrazují se uživateli přesně tak, jak jsou na serveru uloženy. Tento přístup má několik výhod (rychlost, jednoduchost), avšak nehodí se pro vytváření rozsáhlejších projektů. Dynamické webové stránky jsou ty, které mohou v čase měnit svůj obsah. Jedná se o moderní přístup k tvorbě webových stránek, protože umožňuje interakci s uživatelem. Potřebná dynamika je do webových stránek přidávána pomocí tzv. skriptů. Pomocí těchto jazyků jsou vytvářeny tzv. skripty, které jsou kombinovány s jazykem HTML. Podle toho zda je skript proveden na straně serveru, nebo na straně klienta, je možné hovořit o tzv. serverových skriptovacích jazycích a tzv. klientských skriptovacích jazycích. Použití obou typů jazyků je různé a slouží různým účelům.

S rozvojem internetu a informačních technologií (IT) obecně začali také organizace vynakládat značné finanční prostředky na tyto technologie. Podle průzkumů činí objem těchto prostředků v průměru 6,3% jejich důchodu. Tím se informační technologie stávají strategickou zbraní pro současné firmy, které díky nim mají možnost získat konkurenční výhodu ve svém oboru podnikání, zlepšit produktivitu a výkonnost firmy, zavést nové způsoby řízení organizace a vytvořit nové oblasti podnikání. V dnešní době nejde jen o zlepšení pozice firmy na trhu, ale o přežití podniku v novém podnikatelském prostředí. Nutnost nasazení IT vyplývá ze změn, ke kterým dochází uvnitř i vně dané organizace. Z výše uvedeného lze vyvodit, že informační technologie mají zásadní dopad na podnikatelský úspěch firmy, nebo na její fungování.[KALUŽA, 2010]

Hlavním cílem této diplomové práce je navrhnout a vytvořit elektronický rezervační systém hrací plochy zimního stadionu v Litomyšli. Tento cíl bude naplněn za použití moderních nástrojů, postupů a technologií. Primárním motivem pro vytvoření tohoto rezervačního systému je usnadnění procesu rezervace hrací plochy stadionu a zjednodušení správy vytvořených rezervací. Vedlejším motivem je pak zvýšení konkurenceschopnosti a atraktivity mezi zimními stadiony v daném regionu.

2. Teoretické přístupy a nástroje tvorby informačních systémů a internetových prezentací

2.1. HTML a XHTML

HTML (HyperText Markup Language) je značkovací jazyk vytvořený pro systém World Wide Web. Za jeho tvůrce je považován Tim Berners-Lee, který přišel s myšlenkou vytvořit popisný jazyk za použití značek jazyka SGML (Standard Generalized Markup Language). Ten se stal v roce 1986 mezinárodním standardem, přičemž vycházel z pracovní verze SGML z roku 1980. SGML bylo vytvořeno na základech jazyka GML (Generalized Markup Language), který byl vyvinut koncem šedesátých let společností IBM. Jazyk SGML je i dnes aktivním standardem registrovaným organizací ISO (International Organization for Standardization), avšak v praxi se příliš nerozšířil (výjimkou je oblast nakladatelství). Při konstrukci jazyka HTML byl použit také jazyk DTD (Document Type Definition), který sloužil pro definování značek SGML dokumentu. Jazyk DTD má vlastní syntaxi, díky které je možné nadefinovat strukturu pro jiné jazyky. To znamená, definovat jaké značky se budou v jazyku používat, v jakém pořadí a s jakými parametry. [HAUSER, 2006]

Po vytvoření HTML byla nutná také jeho standardizace. Původně byla za jazyk zodpovědná organizace IETF (Internet Engineering Task Force), ale po vzniku HTML 2.0 (1996) byl jazyk přesunut pod organizaci W3C (World Wide Web Consortium), kterou v roce 1994 založil zmiňovaný Tim Berners-Lee. W3C je mezinárodním konsorciem, které vyvíjí standardy pro World Wide Web. Při zpracování návrhů na standardizaci organizací W3C musí každý návrh projít několika stupni schvalování. Tyto stupně jsou následující:

- Working Draft (WD),
- Candidate Recommendation (CR),
- Proposed Recommendation (PR),
- Recommendation (REC).

Na stupni Working Draft se specifikace nachází ve stavu zpracování. Zde se jedná o prvotní materiál, který je připomínkován. Po ukončení diskusí přechází specifikace na další stupeň (Candidate Recommendation), kde se provádí její první implementace a

nasazení v praxi. V následující fázi (Proposed Recommendation) je specifikace znovu a naposledy posuzována. Nejvyšším stupněm je pak Recommendation, kdy je specifikace uznána za platnou a označena novým číslem verze. Poté se již nesmí v této specifikaci nic měnit a všechny další změny musí projít opět celým procesem standardizace. Smyslem této standardizace je usnadnit a sjednotit nasazení technologií v praxi. Ve standardu HTML je uvedeno především to, jak má internetový prohlížeč interpretovat internetovou stránku. Tím se tento standard stává doporučením výrobcům internetových prohlížečů s cílem jednotného zobrazování internetových stránek. [HAUSER, 2006]

Jazyk HTML prošel od svých počátků řadou verzí. V současné době je nejnovějším standardem HTML verze 4.01 z roku 1999. V blízké době se, ale očekává schválení doporučení specifikace HTML 5, který je nyní na stupni schvalování Candidate Recommendation.[W3C - Candidate Recommendation 04 February 2014] I přes tento fakt je již tento standard implementován v řadě internetových prohlížečů a je tak již použitelný pro webové vývojáře. Předchozí verze jsou:

- HTML verze 0.9 – 1.2,
- HTML verze 2.0,
- HTML verze 3.2,
- HTML verze 4.0.

Původně se předpokládalo, že HTML verze 4.01 bude poslední verzí tohoto jazyka a postupně se přejde na novější specifikaci XHTML, která není již postavena na jazyku SGML, ale na jazyku XML (eXtensible Markup Language). Ten byl vydán organizací W3C ve stupni Recommendation v roce 1998. Cílem bylo přizpůsobit jazyk SGML požadavkům současného webu. XML nabízí kromě sady přesně definovaných značek také možnost vytvářet značky vlastní. Z toho důvodu se nejedná o popisný jazyk, nýbrž o tzv. metajazyk. To znamená, že je možné pomocí jazyka XML definovat další jazyky na něm založené. K tomuto účelu je možné použít zmíněného jazyka DTD, nebo XML schémat. Obě technologie umožňují definovat strukturu vytvářeného jazyka. Je tedy pevně stanoveno jaké značky a parametry je možné používat. Tímto způsobem byl také organizací W3C definován jazyk XHTML. [HAUSER, 2006]

XHTML (eXtensible Hypertext Markup Language), je tedy značkovací jazyk pro systém world wide web, stejně jako HTML. Jak již bylo řečeno, původní předpoklad

byl, že tento novější jazyk plně nahradí HTML. První verze tohoto jazyka byla vydána v roce 2000 formou doporučení organizace W3C. Postupem času vznikly další verze XHTML:

- XHTML 1.0,
 - XHTML 1.0 Strict,
 - XHTML 1.0 Transitional,
 - XHTML 1.0 Frameset,
- XHTML 1.1
- XHTML 2.0

Poslední verze XHTML, tedy verze 2.0, nebyla nikdy vydána ve stupni Recommendation. Ačkoliv byla roku 2007 vytvořena pracovní skupina XHTML 2.0, ukázalo se být tato specifikace slepou uličkou. Roku 2009 rozhodla organizace W3C o rozpuštění pracovní skupiny XHTML 2.0 s tím, že uvolněné zdroje budou použity k urychlení vývoje specifikace HTML 5. Na konci roku 2010 byla tato pracovní skupina opravdu uzavřena a jedinou specifikací, na které se pracuje v souvislosti s jazyky HTML a XHTML je specifikace HTML 5.[W3C - XHTML2 Working Group Home Page]

2.1.1. HTML 5

HTML 5 je tedy nejnovější specifikací značkovacího jazyka HTML. Jak již bylo řečeno, nyní se nachází ve stadiu schvalování Candidate Recommendation a tedy se stále nejedná o finální specifikaci. Nejnovější verze tohoto dokumentu (CR) byla vydána organizací W3C v únoru 2014.[W3C - Candidate Recommendation 04 February 2014]

HTML 5 tvoří základ pro příští generaci webových aplikací. Pomocí této technologie je možné tvořit weby, které se snadněji vyvíjí a spravují a jsou přívětivější k uživatelům. Tato specifikace jazyka obsahuje nové elementy pro definici struktury a vkládání obsahu, což vede k tomu, že není nutné používat další doplňky. HTML 5 přináší mnohá další vylepšení jako například:

- popisnější zdrojový kód,
- multimedia s menší závislostí na zásuvných modulech,
- lepší aplikace,
- posílání zpráv mezi dokumenty,
- webové sockety,

- úložiště na straně klienta,
- lepší uživatelské rozhraní,
- lepší formuláře,
- vylepšená přístupnost.

Dalším důvodem pro použití HTML 5 již v této vývojové fázi je jeho zpětná kompatibilita. To znamená, že stránka napsaná v HTML 5 se zobrazí i ve starších prohlížečích a také úspěšně projde validací ve validátoru W3C (s upozorněním na použití experimentální technologie). Samozřejmě nelze očekávat dostupnost nových elementů HTML 5, které daný prohlížeč nepodporuje.[HOGAN, 2011]

Ve specifikaci HTML 5 je zavedeno mnoho nových elementů, na druhou stranu jsou odstraněny elementy zastaralé, viz Tab. 2.1. Dále je zde k dispozici mnoho nových atributů a hodnot atributů jako například hodnoty atributu type elementu input, kterými jsou: color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url, week. Pomocí těchto hodnot je možné zobrazovat nové formulářové prvky. Kompletní přehled nových elementů, atributů a jejich hodnot je k dispozici na webu w3schools.[w3schools - HTML5 New Elements]

Nové elementy	Odstraněné elementy
<article>	<acronym>
<aside>	<applet>
<audio>	<basefont>
<bdi>	<big>
<canvas>	<center>
<datalist>	<dir>
<details>	
<dialog>	<frame>
<embed>	<frameset>
<figcaption>	<noframes>
<figure>	<strike>
<footer>	<tt>
<header>	
<keygen>	
<main>	

<mark>	
<meter>	
<nav>	
<output>	
<progress>	
<rp>	
<rt>	
<ruby>	
<section>	
<source>	
<summary>	
<time>	
<track>	
<video>	
<wbr>	

Tab. 2.1 Nové a odstraněné značky v HTML 5

Nové elementy je možné použít například k definování struktury stránky. Tyto tzv. strukturální značky zde mohou nahradit dříve používaný neutrální element div, viz Obr. 2.1. Na tomto místě je také dobré zmínit, jakým způsobem se deklaruje typ dokumentu tzv. DOCTYPE v HTML 5. Způsob zápisu je následující: **<!DOCTYPE html>**. V porovnání s deklarací například XHTML 1.0. Transitional je tento zápis mnohem jednodušší. Deklarace XHTML vypadá následovně:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```



Obr. 2.1 Struktura stránky pomocí nových značek zdroj:[w3schools - HTML5 Semantic Elements]

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4 <header>
5 <h1>Praesent sagittis sit amet urna a dignissim.</h1>
6 </header>
7 <nav>
8 <a href="/html/">HTML</a> |
9 <a href="/css/">CSS</a> |
10 <a href="/js/">JavaScript</a> |
11 <a href="/jquery/">jQuery</a>
12 </nav>
13 <article>
14 <p>In at pharetra tellus. Fusce semper, leo vitae tristique sodales, leo eros commodo libero.</p>
15 </article>
16 <footer>
17 <p>Nunc quis felis lectus.</p>
18 </footer>
19 </body>
20 </html>

```

Obr. 2.2 Ukázka kódu v HTML 5

2.2. CSS

CSS (Cascading Style Sheets), neboli kaskádové styly je v podstatě jazyk popisující způsob zobrazení dokumentů napsaných v jazyku HTML. Hlavním motivem pro používání CSS je oddělení vzhledu stránky od jejího obsahu. Pomocí jazyka HTML se tedy vytváří obsah stránky a pomocí CSS je definován její vzhled. Při potřebě změny vzhledu dokumentu tedy odpadá nutnost změny jeho struktury. [HAUSER, 2006]

První verze CSS byla uvedena v roce 1996 organizací W3C doporučením (REC) pod názvem Cascading Style Sheets, level 1. V roce 1998 bylo vydáno doporučení pro specifikaci Cascading Style Sheets, level 2. Tato verze přinášela určitá vylepšení jako například výběr zobrazovacího média, nebo možnost pozicování elementů. Nejnovější verzí je pak specifikace Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) vydaná doporučením v roce 2011. Zatím nedokončeným standardem je specifikace CSS Level 3. [W3C - Cascading Style Sheets (CSS) Snapshot 2010]

Největším přínosem použití kaskádových stylů je možnost změny definice zobrazení jednotlivých elementů na jediném místě. Taková změna se pak projeví ve všech prvcích, které daný styl používají. Kaskádové styly jsou tedy jednoduchým nástrojem k aktualizaci formátování dokumentu a udržení jeho konzistentnosti. Jednotlivé styly je možné zformovat do šablon, což umožňuje na jeden dokument aplikovat více stylů. Existuje několik způsobů (viz Obr. 2.3), kterými je možné aplikovat kaskádové styly na HTML dokument:

- inline zápis s použitím atributu style u formátovaného elementu,
- použití elementu style v hlavičce stránky,
- použití externího CSS souboru (viz Obr. 2.4), na který je v hlavičce odkazováno pomocí elementu link.

Při použití elementu style v hlavičce stránky je nutné, v atributu type, uvést příslušný MIME (Multipurpose Internet Mail Extensions) typ, ve většině případů je zde uvedeno text/css. V případě použití elementu link v hlavičce dokumentu se používají tři atributy. Atribut rel obsahuje v případě připojení CSS souboru hodnotu stylesheet a vyjadřuje vztah mezi dokumentem a odkazovaným dokumentem. Atributem type je opět specifikován MIME typ odkazovaného souboru a atributem href je specifikováno umístění odkazovaného souboru.[SCHAFFER, 2009]

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3   <head>
4     <title></title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6     <style type="text/css">
7       h1 {color: red;}
8       h2 {background-color: blue;}
9     </style>
10  </head>
11  <body>
12    <p style="color: yellow;">Žlutý odstavec</p>
13    <div id="left">
14      <div id="left_in">
15        <ul class="menu">
16          <li class="item"><a href="index.php">Home</a></li>
17          <li class="item"><a href="kontakt.php">Kontakt</a></li>
18          <li class="item"><a href="cenik.php">Ceník</a></li>
19        </ul>
20      </div>
21    </div>
22  </body>
23 </html>

```

Obr. 2.3 Způsoby zápisu kaskádových stylů

Všechny uvedené definice kaskádových stylů mají stejný formát zápisu. V každé definici je nejprve deklarován určitý selektor, který reprezentuje konkrétní element v HTML dokumentu. Nejjednodušším názvem selektoru je přímo název daného elementu (např. h1). Ve složitější podobě může selektor obsahovat názvy dceřiných elementů (viz Obr. 2.4, řádek 14). Dále obsahuje každá definice pár složených závorek, mezi nimiž je umístěn výčet jedné, nebo více vlastností a jejich

hodnot. Za každým párem vlastnost a její hodnota je umístěn středník. Pokud je potřeba odlišit od sebe jednotlivé shodné elementy v rámci jednoho dokumentu (např. každý odstavec má jiný vzhled), je možné použít tzv. třídy. Třída je tvořena názvem příslušného elementu, za kterým následuje tečka a název třídy. Pokud se má definovaná třída použít, přidá se danému elementu atribut class obsahující název požadované třídy (viz. Obr. 2.3.). Podobně jako třídy fungují také identifikátory s tím rozdílem, že v jednom dokumentu by se měl vyskytovat pouze jeden element s určitým identifikátorem. Identifikátor je tvořen názvem příslušného elementu, za kterým následuje mřížka a název identifikátoru. V HTML dokumentu se pak identifikátor používá stejně jako třída, pouze atribut class je nahrazen atributem id.

```
1 #left
2 {
3 float: left;
4 width: 200px;
5 min-height:800px;
6 }
7 #left_in
8 {
9 font-family:Arial, Helvetica, sans-serif;
10 font-size: 14px;
11 padding:5px 8px 5px 8px;
12 background-image: url('imgs/background.png');
13 }
14 ul.menu li.item a
15 {
16 display:block;
17 width: 200px;
18 height: 30px;
19 line-height: 30px;
20 text-align: left;
21 text-indent:40px;
22 color: #52b4e5;
23 font-size: 20px;
24 text-decoration: none;
25 }
26 ul.menu li.item a:hover
27 {
28 font-weight:bold;
29 }
```

Obr 2.4 Zápis CSS v externím souboru

2.2.1. CSS 3

Jak již bylo uvedeno, CSS 3 je nejnovější specifikací kaskádových stylů, která je ale ovšem stále ve stádiu vývoje. Přesto se již hojně používá a to převážně v kombinaci

s HTML 5. Spolu s HTML 5 je také specifikace CSS 3 více či méně implementována moderními internetovými prohlížeči. Tyto dva nové standardy, představené organizací W3C, jsou tak novou generací používaných technologií při tvorbě moderních webových aplikací. Důležitý je přitom fakt, že standard CSS 3 je plně kompatibilní s předchozími verzemi. CSS 3 se skládá z tzv. modulů. Tyto moduly obsahují dřívější specifikaci CSS rozdělenou na menší části a jsou rozšířeny o nové moduly. Některé z nejdůležitějších modulů jsou:

- selektory,
- box model,
- pozadí a ohraničení,
- obrázkové hodnoty a náhrada obsahu,
- textové efekty,
- 2D/3D transformace,
- animace,
- vícesloupcový layout,
- uživatelské prostředí.

Nově přidané moduly nabízejí mnoho nových vlastností, které umožňují zcela nové použití kaskádových stylů. Nově je možné například vytvářet kulaté rohy, přidávat stíny boxům, používat obrázky jako ohraničení, používat několik obrázků jako pozadí, vytvářet hladké přechody mezi dvěma, nebo více barvami, přidávat stíny textu, používat fonty, které nejsou nainstalovány na uživatelské pc, používat 2D transformace (posun, rotace, změna velikosti, zešikmení), používat 3D transformace (rotace), vytvářet animace, vytvářet více textových sloupců atd. Tyto vlastnosti posouvají použití kaskádových stylů na novou úroveň.[w3schools - CSS3 Introduction]

2.3. JavaScript

JavaScript je multiplatformní skriptovací jazyk, který byl původně vyvinut za účelem ověřování údajů z webových formulářů na straně klienta. Před nástupem této technologie bylo pro ověření těchto údajů nutné používat technologie na straně serveru. Taková validace byla, ale značně časově náročná, vzhledem k tehdejší rychlosti připojení k internetu. Postupem času se z jazyku JavaScript stala nepostradatelná součást webových prohlížečů, která se stará o interakci s prvky okna

prohlížeče a jeho obsahu. JavaScript je plnohodnotný programovací jazyk schopný provádět složité výpočty a interakce.[ZAKAS, 2009]

Na začátku 90. let 20. století zahájila společnost Nombas vývoj vloženého skriptovacího jazyka nazvaného C-minus-minus (Cmm). Cílem bylo vytvořit skriptovací jazyk, který by nahradil makra, ale zůstal by podobný jazyku C. Jazyk Cmm byl dodáván v produktu s názvem CEnvi. Jméno tohoto jazyka bylo nakonec společností změněno na ScriptEase. Později byla vyvinuta verze produktu CEnvi, kterou bylo možno do webových stránek. Tato verze byla pojmenována Espresso Pages a lze ji považovat za první skriptovací jazyk na straně klienta použitý na webu. S rostoucí popularitou internetu začala také společnost Netscape uvažovat o vlastním vývoji skriptovacího jazyka na straně klienta, který by implementovala do svého webového prohlížeče Netscape Navigátor. Ten se tehdy těšil obrovské oblibě. Proto Brendan Eich (společnost Netscape) začal pracovat na vývoji jazyka LiveScript pro Netscape Navigator 2, který měl být vydán v roce 1995. Vzhledem k potřebě včasného dokončení přizvala společnost Netscape ke spolupráci společnost Sun microsystems (autor jazyku java). Před vydáním v prohlížeči Netscape Navigator 2 byl jazyk LiveScript přejmenován na JavaScript a to především z marketingových důvodů. Díky úspěchu JavaScriptu byla pro další verzi Netscape Navigatoru vyvinuta jeho nová verze 1.1. V té době se také společnost Microsoft rozhodla implementovat JavaScript ve svém prohlížeči Internet Explorer 3. Tato implementace byla pojmenována JScript. To vedlo k tomu, že současně vedle sebe existovaly 3 různé implementace jedné technologie (CEnvi, JavaScript, JScript) a bylo tedy zapotřebí vytvořit určité standardy. [ZAKAS, 2009]

V roce 1997 byl předán JavaScript 1.1. organizaci ECMA (European Computer Manufacturers Association) jako návrh pro standardizaci. Byla vytvořena komise z programátorů společností Netscape, Sun, Microsoft, a dalších, která vytvořila standard ECMA-262. Tento standard definuje nový jazyk nazvaný ECMAScript. Termíny JavaScript a ECMAScript se často zaměňují, ale JavaScript je mnohem obecnější termín. Zahrnuje totiž 3 odlišné části: [ZAKAS, 2009]

- jádro (ECMAScript),
- objektový model dokumentu (DOM),
- objektový model prohlížeče (BOM).

2.3.1. ECMAScript

Jazyk ECMAScript není svázán s webovými prohlížeči a neobsahuje žádné metody pro vstup a výstup. Tento standard je definován jako základ pro budování rozsáhlejších skriptovacích jazyků. Webový prohlížeč je zde pouze jedním z možných hostitelských prostředí. Hostitelské prostředí poskytuje implementaci standardu ECMAScript a rozšíření pro komunikaci s tímto prostředím. Standard ECMAScript-262 tedy specifikuje tyto části jazyka: [ZAKAS, 2009]

- syntaxe,
- typy,
- příkazy,
- klíčová slova,
- vyhrazená slova,
- operátory
- objekty.

Verze ECMAScriptu jsou definovány jako jednotlivé edice. Nejnovější edicí tohoto standardu je edice 5.1 vydaná v květnu 2011, která je plně v souladu s třetí edicí mezinárodního standardu ISO / IEC 16262:2011. [<http://www.ecma-international.org/>] Tato edice je v současnosti podporována ve všech nejnovějších verzích nejpoužívanějších webových prohlížečů (Chrome, Firefox, Internet Explorer, Opera, Safari).

2.3.2. Objektový model dokumentu (DOM)

Objektový model dokumentu (Document object model) je aplikační programovací rozhraní pro jazyk XML a HTML. DOM mapuje prvky dokumentu do hierarchie uzlů a umožňuje přístup k jejich editaci. Původně měli Internet Explorer a Netscape Navigator vlastní specifické rozhraní pro manipulaci s prvky HTML dokumentu. To sice umožňovalo práci se vzhledem a obsahem webových stránek pomocí skriptovacího jazyku na straně klienta, ale působilo to nekompatibilitu mezi těmito webovými prohlížeči. Bylo tedy rozhodnuto o standardizaci s cílem zachovat platformní nezávislost webu. Z toho důvodu vytvořila organizace W3C Document Object model. První specifikace byla vydána doporučením (REC) v roce 1998. Jednalo se o tzv. DOM level 1, který se skládal ze dvou modulů. Prvním byl DOM Core pomocí něhož bylo

zajištěno mapování prvků XML dokumentů a druhým byl DOM HTML, rozšiřující DOM core o specifika jazyku HTML. Postupně následovaly další verze: [ZAKAS, 2009]

- Document Object Model Level 2
- Document Object Model Level 3
- DOM4

Poslední verze DOM 4 je stále ve stádiu schvalování Working Draft, z čehož vyplývá, že nejnovější vydané doporučení je Document Object Model Level 3, vydané v roce 2004. [W3C - Document Object Model (DOM) Technical Reports]

2.3.3. Objektový model prohlížeče (BOM)

Objektový model prohlížeče (Browser Object Model) umožňuje interakci s oknem webového prohlížeče mimo kontext zobrazené stránky. Obecně lze definovat BOM jako rozšíření JavaScriptu specifické pro webový prohlížeč. Takovým rozšířením je například:

- otevírání nového okna prohlížeče,
- přesouvání, změna velikosti a zavírání okna prohlížeče,
- objekt navigator, poskytující informace o prohlížeči,
- objekt location, poskytující údaje o stránce načtené v okně prohlížeče,
- objekt screen , poskytující informace o rozlišení obrazovky
- podpora pro cookies
- vlastní objekty (XMLHttpRequest, ActiveXObject)

BOM tvoří jedinou část JavaScriptu, která není definována žádným standardem. Z toho vyplývá, že každý prohlížeč obsahuje svou vlastní implementaci této technologie. [ZAKAS, 2009]

2.3.4. JavaScript + HTML

Zavedením jazyka JavaScript na pole webových stránek vznikla potřeba jeho propojení s jazykem HTML, který je v této oblasti dominantní. Společnost Netscape se již v rámci prací na jazyku JavaScript zabývala jeho integrací do HTML stránky tak, aby nenarušil správné zobrazování stránek v ostatních prohlížečích. Výsledkem byla dohoda o univerzální podpoře skriptování na webových stránkách. Hlavní metodou pro vložení skriptu do HTML stránky se stalo použití elementu script. Tento element se stal

součástí formální specifikace jazyka HTML. V rámci specifikace HTML 4.01 jsou elementu script definovány tyto atributy:

- charset (nepovinný) – specifikace znakové sady kódu,
- defer (nepovinný) – označuje fakt, že skript bude proveden až po analýze a kompletním zobrazení dokumentu,
- language (zastaralý) – původní označení skriptovacího jazyka,
- src (nepovinný) – odkazuje na externí soubor obsahující kód, který se má provést,
- type (povinný) – nahrazuje atribut language, specifikuje MIME typ skriptovacího jazyka (např. text/javascript),

Vložení skriptu pomocí elementu script je možné dvěma způsoby. Prvním způsobem je vložení skriptu přímo do HTML stránky a druhým je vložení skriptu z externího souboru.[ZAKAS, 2009] Existuje také varianta bez použití elementu script a to inline zápis kódu do atributu události jiného elementu (např. událost onClick). Použití všech variant viz Obr. 2.5

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3   <head>
4     <title></title>
5     <script type="text/javascript" src="skript.js"></script>
6   </head>
7   <body>
8     <script type="text/javascript">
9       function den() {
10         den = window.prompt('Co je dnes za den', 'Pondělí');
11         if (den) window.alert("Dnes je "+den);
12         else window.alert("Den nebyl zadán");
13       }
14     </script>
15     <input type="button" onClick="den();" value="Zadat den" />
16   </body>
17 </html>
```

Obr. 2.5 Zápis JavaScriptu

2.4. jQuery

jQuery je knihovna založená na jazyku JavaScript, přičemž důraz je kladen na interakci mezi ním a jazykem HTML. Jedná se o velmi rychlou a lehkou knihovnu, která zjednodušuje tvorbu a správu událostí, animací a dalších komponent. V současnosti jsou trendem dynamické internetové stránky, potřebné dynamiky je možné dosáhnout nasazením jazyka JavaScript, jehož použití, ale není zcela triviální záležitostí. Tento fakt vedl k vytvoření knihovny jQuery, díky níž je použití jazyka JavaScript velmi usnadněno. Tato knihovna byla představena v roce 2006 Johnem Resigem, jehož cílem bylo jednoduché vyhledávání elementů v dokumentu a práce s nimi. Dnes je knihovna šířena pod licencí MIT (Massachusetts Institute of Technology), což znamená, že je volně použitelná na libovolném webovém projektu, včetně komerčního využití. V knihovně jQuery je obsaženo mnoho funkcí využívaných při vývoji webových aplikací, přičemž její hlavní síla spočívá především ve výběru konkrétních elementů v rámci modelu DOM. Elementy jsou vybírány pomocí selektorového jádra Sizzle. Dále je umožněno tyto vybrané elementy modifikovat, např. je možné měnit CSS styly daných elementů. Také je možné pomocí knihovny jQuery zpracovávat události (např. `onMouse()`). Důležitou částí knihovny je také široká nabídka efektů, jejichž použití je velmi jednoduché a účinné napříč internetovými prohlížeči. Pomocí jQuery je také umožněno vytvářet animace, díky čemuž je možné se vyhnout nasazení složitějších technologií. V neposlední řadě je zde nabízena spolupráce knihovny jQuery s technologií AJAX (Asynchronous JavaScript and XML) a výrazně tak ulehčuje její používání. Aby byla zachována co možná nejmenší velikost knihovny (komprimována 94kB), jsou využívány tzv. zásuvné moduly. Díky nim je umožněno použití ucelených balíčků hotových řešení (např. validace formulářů, obrázková galerie atd.). V rámci knihovny jQuery je také možné vytvářet vlastní zásuvné moduly.[MARGORÍN, 2011]

Implementace knihovny jQuery do HTML dokumentu je velmi jednoduchá, přičemž existují dva různé způsoby jak jí docílit. Prvním způsobem je stažení knihovny z oficiálních stránek jQuery (<http://www.jquery.com/>). Celá knihovna je distribuována v rámci jednoho JavaScript souboru. Tento soubor je pak standardně připojen v rámci hlavičky HTML dokumentu. Druhou možností jak připojit knihovnu jQuery je pak CDN (Content Delivery Network) viz Obr. 2.6. Některé společnosti (Microsoft, Google) poskytují knihovnu volně, právě prostřednictvím svých sítí CDN. Soubor knihovny

jQuery je umístěn na výkonných serverech, které jsou rozmístěné po celém světě. Tím je docíleno toho, že uživatelé mají zajištěn rychlý přístup odkudkoliv. Použití sítě CDN může, díky distribuovaným serverům a využití mezipaměti webového prohlížeče, urychlit stahování pro koncové uživatele. Pro vývojovou fázi projektu je pravděpodobně praktičtější lokální umístění souboru a to z důvodu nezávislosti na internetovém připojení. [CHAFFER, 2013]

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3   <head>
4     <title></title>
5     <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
6     <script type="text/javascript">
7       $(document).ready(function() {
8         $(".button").click(function() {
9           $("#main").slideToggle("slow");
10          return false;
11        });
12      });
13    </script>
14    <link rel="stylesheet" type="text/css" href="styly.css">
15  </head>
16  <body>
17    <div id="main">
18    </div>
19    <p class="slide">
20      <a href="#" class="button">Tlačítko</a>
21    </p>
22  </body>
23 </html>
```

Obr. 2.6 Použití knihovny jQuery

2.5. PHP

PHP (PHP: Hypertext Preprocessor) je serverový skriptovací jazyk, který slouží k vývoji dynamických webových aplikací. Za vývojem PHP nestojí žádná organizace, nýbrž jednotlivci, kteří udržují tento jazyk jako technologii s otevřenými zdrojovými kódy (open source). To znamená, že spolu s jazykem je k dispozici i jeho zdrojový kód, což přináší možnost úprav a zlepšování komukoliv, kdo je těchto úprav schopen. Díky otevřenému zdrojovému kódu je také možná širší kontrola, což může vést k menší chybovosti a větší bezpečnosti této technologie. Jazyk PHP vyniká svou jednoduchostí a fungováním napříč různými operačními systémy a webovými servery. Tímto se tento jazyk stává velmi silným nástrojem pro tvorbu dynamických webových stránek.

Historie jazyka PHP se začala psát kolem roku 1995, kdy dánský programátor Rasmus Lerdorf vydal sadu skriptů s PHP/FI, napsaných v jazyku Perl. Díky velké

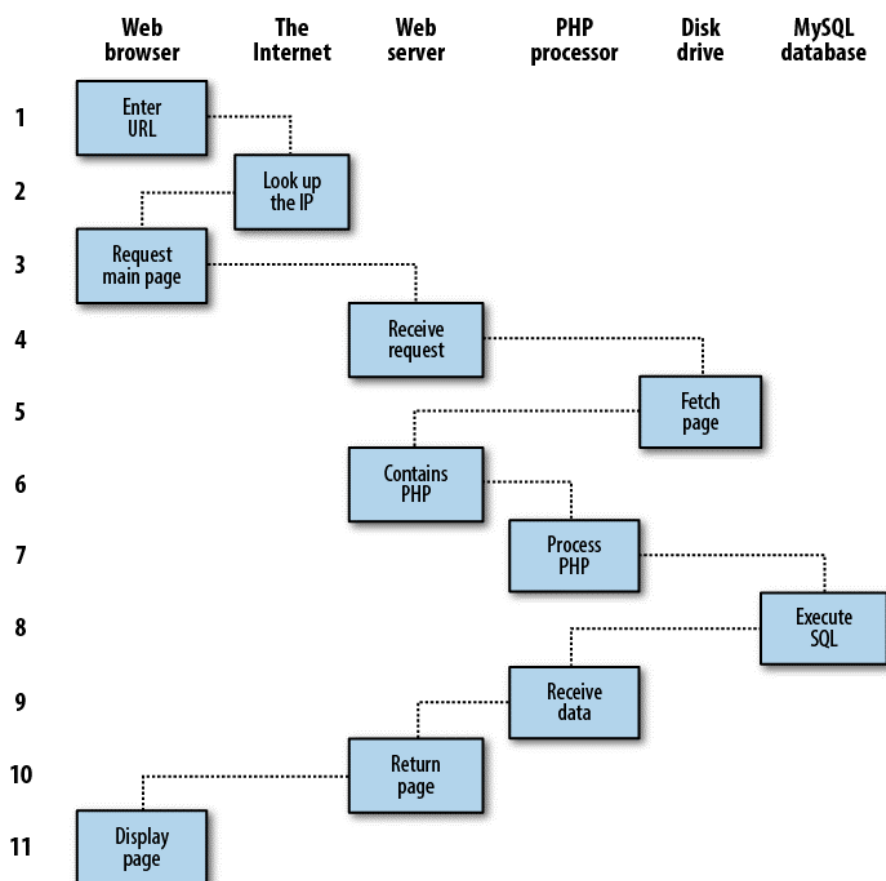
používanosti této sady vznikla v roce 1997 druhá verze PHP/FI, která je již napsána v jazyku C. Téhož roku se pokoušeli Andi Gutmans a Zeev Suraski použít tuto sadu pro vývoj komerčních aplikací, přičemž usoudili, že je pro tyto účely poddimenzovaná. Rozhodli se tyto skripty kompletně přepsat a výsledek označili názvem PHP 3. Za účelem zvýšení výkonu pro složité aplikace bylo v roce 1998 přepsáno jádro PHP. Nové jádro bylo nazváno Zend (složení jmen tvůrců) a v roce 2000 bylo vytvořeno PHP verze 4, postavené na tomto jádru. Po dalším vylepšení Zend jádra bylo vydáno PHP verze 5 (rok 2004). Hlavním přínosem této verze je podstatně vylepšená podpora objektově orientovaného programování. V současnosti je PHP verze 5 aktuální verzí (konkrétně PHP5.5). [PONKRÁC, 2007]

Programovací jazyk PHP byl vyvinut za účelem obohacení statických webových stránek o dynamické skriptování na straně serveru. Dříve se za tímto účelem používaly programy, které byly volány přes Common Gateway Interface (CGI). Pro spuštění těchto programů musel webový server spustit nový proces, který byl pro operační systém velmi náročný a vyžadoval pro své zpracování dlouhý čas. To bylo značně nevýhodné zvláště u velmi frekventovaných webových stránek. S příchodem PHP došlo ke značnému zrychlení vyřízení požadavků uživatele a to zejména díky jinému principu fungování. Funkce PHP jsou načítány přímo jako součást webového serveru a ke spuštění jsou přiváděny prostřednictvím procesu webového serveru. Každý soubor s příponou php je tedy zpracován interpretem jazyka PHP. V rámci interpreteru je soubor prověřen a jsou provedeny zadané instrukce. Poté je výsledek předán zpět na webový server a odtud je daný obsah poslán klientovi (nejčastěji internetovému prohlížeči). Mezi nejpodstatnější instrukce jazyka PHP můžeme zařadit značky ohraničující PHP kód. Pokud nejsou tyto značky nalezeny interpretem jazyka PHP v rámci souboru, je vrácen zpět text zdrojového kódu, aniž by se jakékoliv instrukce vykonaly. Existuje několik zápisů těchto značek, přičemž pouze dva z nich fungují vždy. Jedním zápisem je: `<?php ?>` (viz Obr.2.8) a druhým `<script language="php"></script>`. [KOFLER, 2007]

Zmíněný proces zpracování PHP souboru je možno detailně rozpracovat do jedenácti kroků (viz Obr.2.7). Tímto postupem je naznačeno zpracování dynamické webové stránky od jejího volání až po její zobrazení: [NIXON, 2012])

1. uživatel zadá požadovanou URL (Uniform Resource Locator) adresu (např. <http://www.server.com>) do adresního řádku internetového prohlížeče,

2. internetový prohlížeč vyhledá IP (Internet Protocol) adresu požadovaného serveru,
3. internetový prohlížeč zašle na tuto adresu požadavek o domovskou stránku daného serveru,
4. vyslaný požadavek dorazí prostřednictvím internetu k cílovému webovému serveru,
5. webový server, který přijal daný požadavek, načte domovskou stránku ze svého pevného disku,
6. webový server, který má nyní stránku načtenou ve své paměti, zjistí, že daný soubor obsahuje PHP skript a předá jej PHP interpreteru,
7. PHP interpreter vykonává daný skript,
8. pokud část PHP kódu obsahuje MySQL příkazy, předají se tyto MySQL databázovému jádru,
9. MySQL databáze vrátí výsledky zpracování těchto příkazů zpět PHP interpreteru,
10. PHP interpreter vrátí výsledky vykonaného PHP skriptu webovému serveru,
11. Webový server vrátí webovou stránku danému klientovi, který ji zobrazí.



Obr. 2.7 Proces zpracování požadavků zdroj: [NIXON, 2012]

Jak je již patrné z výše uvedeného, PHP skripty jsou zapisovány přímo do HTML dokumentu a to mezi uvedené značky. Každá část PHP kódu je složena z jednotlivých příkazů, přičemž každý jednotlivý příkaz je oddělen středníkem (viz Obr. 2.8). Příkazy jsou pak vykonávány v pořadí, ve kterém jsou zapsány.

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3   <head>
4     <title></title>
5   </head>
6   <body>
7     <?php
8       $hour = date("G");
9       if ($hour < 9 && $hour >=4)
10         echo "Je ráno";
11       else if ($hour < 12 && $hour >= 9)
12         echo "Je dopoledne";
13       else if ($hour == 12)
14         echo "Je poledne";
15       else if ($hour < 18 && $hour > 12)
16         echo "Je odpoledne";
17       else if ($hour < 22 && $hour >=18)
18         echo "Je večer";
19       else
20         echo "Je noc";
21     ?>
22   </body>
23 </html>

```

Obr. 2.8 zápis PHP kódu

2.6. MySQL

Jen velmi malé procento dynamických webových aplikací dnes funguje bez připojení k nějaké databázi. Databáze je ve svém původním významu místo, ve kterém jsou uložena data. Data v databázi jsou spravována tzv. systémem řízení báze dat (DataBase Management Systém - DBMS). Pomocí DBMS je prováděno veškeré získávání, zpracování a ukládání dat v rámci databáze. Mezi DBMS patří celá řada programů jak komerčních tak nekomerčních produktů. Příkladem nekomerčního DBMS může být právě MySQL, je však pravdou, že v některých případech je vyžadováno použití placené podnikové verze. Naprostá většina moderních DBMS je postavena na tzv. relačním modelu. Tento model je založen na databázových tabulkách (relacích), přičemž každá databázová tabulka sdružuje data určitého druhu. Záznamy v tabulce jsou strukturovány do tzv. sloupců, přičemž každý sloupec je definován unikátním názvem, typem a rozsahem. Mezi jednotlivými tabulkami jsou pak definovány vztahy. Většina databází pracuje na architektuře klient/server. Tento princip byl podrobně popsán v předchozí kapitole (viz Obr. 2.7). Systém řízení báze dat je v tomto případě součástí webového serveru a pomocí něj je přístupováno

k datům v databázi. Prostřednictvím DBMS je tedy možná kompletní správa dat v rámci dané databáze.[PONKRÁC, 2007]

MySQL je databázový systém (DBMS + databáze) vytvořený soukromou švédskou firmou MySQL AB, která se později stala součástí společnosti Oracle. Ve srovnání s konkurenčními produkty patří MySQL spíše k těm jednodušším. Na druhou stranu se jedná o malý, rychlý a nenáročný databázový systém. Nejčastěji je nasazován spolu s technologií PHP, o které již byla řeč. Pak tedy skripty napsané v jazyku PHP mohou obsahovat příkazy pro práci s daty v dané databázi. Tyto příkazy jsou následně předány k provedení systému MySQL, přičemž výsledky těchto operací jsou předány zpět PHP interpretu (viz Obr. 2.7). Jakákoliv práce s daty v rámci dané databáze je prováděna prostřednictvím následujících operací:

- Navázání spojení s databázovým systémem,
- Zaslání příkazů k práci s danými daty,
- Obdržení výsledných dat,
- Ukončení spojení s databázovým systémem.

Komunikace mezi PHP skriptem a systémem MySQL probíhá prostřednictvím speciálního jazyka SQL (Structured Query Language). Jazyk SQL byl poprvé implementován společností IBM v roce 1974. V roce 1986 byl tento jazyk standardizován společností ANSI (American National Standards Institute) a od té doby se stal univerzálním nástrojem pro práci nad daty relačních databází.[PONKRÁC, 2007]

Pomocí jazyka SQL je možné provádět příkazy nad databází, přijímat data z databáze, vkládat záznamy do databáze, upravovat záznamy v databázi, odstraňovat záznamy z databáze, vytvářet nové databáze, vytvářet nové databázové tabulky, vytvářet uložené procedury v databázi, vytvářet pohledy nad databází a nastavovat oprávnění pro tabulky, procedury a pohledy. Těchto možností je dosaženo prostřednictvím příkazů, z nichž nejpoužívanější jsou:[w3schools - Introduction to SQL]

- SELECT – pro výběr dat,
- UPDATE – pro modifikaci dat,
- DELETE – pro odstranění dat,
- INSERT INTO – pro vkládání nových dat,

- CREATE DATABASE – pro vytváření nových databází,
- ALTER DATABASE – pro modifikaci databází,
- CREATE TABLE – pro vytváření nových tabulek,
- ALTER TABLE – pro modifikaci tabulek,
- DROP TABLE – pro odstranění tabulek,
- CREATE INDEX – pro vytváření indexů
- DROP INDEX – pro odstranění indexů.

2.7. Projektování informačních systémů

Informační systém lze podle Encyclopædia Britannica definovat jako: „*Integrovanou množinu komponent pro sběr ukládání zpracování a předávání informací znalostí a digitálních produktů*“.[Encyclopædia Britannica] O informačních systémech se začalo intenzivně hovořit až po nástupu informačních technologií. Dnes jsou oba termíny tak pevně spjaté, že je téměř nemožné představit si jakýkoliv informační systém, který by existoval bez aplikace informačních technologií. Nejedná se tedy o jednu technologii, ale o jejich soubor.[KALUŽA, 2010]

Plánování informačních systémů je prováděno prostřednictvím projektů. V případě jednoduchých informačních systémů je možné jejich projektování prostřednictvím jednoho projektu. V případě složitějších systémů je však nutné použití více projektů, kterými mohou být: části systému, etapy vývoje systému, kombinace obou možností. [KALUŽA, 2010]

V předchozím odstavci byl zaveden termín etapa vývoje informačního systému. Množinu těchto vývojových etap je možné označit jako životní cyklus vývoje informačního systému. Strukturu jednotlivých etap v rámci jednoho životního cyklu definuje Kaluža [2012, s. 10] takto:

1. „*identifikace a výběr projektů – zde se identifikuje potřeba nového systému, požadavky uživatelů, podnikatelské cíle, priority vývoje systému,*
2. *zahájení a plánování projektů – v této etapě se přesně obsahově vymezí jednotlivé projekty, stanoví se řešitelský tým, určí se potřebné zdroje a časový*

plán dalšího postupu, vedení organizace učiní konečné rozhodnutí o realizaci projektů,

- 3. analýza současného stavu – současný informační systém je kriticky analyzován, hledají se jeho nedostatky a možnosti zdokonalení, poznatky z analýzy se konfrontují s požadavky uživatelů, analyzuje se stav využití informačních technologií, formulují se alternativy návrhu nového řešení,*
- 4. návrh (projektování) nového řešení – tvorba nového systému, navrhuje se datové struktury, vstupní formuláře a výstupní sestavy, dialogy, algoritmy, struktura programového řešení; někdy se rozlišuje mezi logickým a fyzickým návrhem – logický návrh systému je nezávislý na technologické platformě, fyzický návrh systému je těsně svázán s programovým a hardwarovým řešením,*
- 5. zavedení (implementace) nového řešení – testování programů, instalace konečné verze softwaru, finalizace dokumentace, školení uživatelů, zkušební provoz systému, předání k rutinnímu provozu,*
- 6. údržba systému – úpravy systému v důsledku změny vnějších podmínek, odstraňování skrytých závad, zlepšování funkcí systému.,,*

2.7.1. Identifikace a výběr projektů

Tato etapa je první etapou životního cyklu vývoje informačních systémů (IS). V této etapě lze realizovat v následujících krocích:

- identifikace možných projektů vývoje IS,
- posouzení navržených projektů,
- výběr projektů pro realizaci.

Náměty na racionalizaci systému se získávají tzv. postupem shora dolů. V rámci tohoto postupu je zajištěno získávání informací začínající u vedení společnosti. Dalším možným postupem je tzv. zdola nahoru. V tomto případě je zajištěno získávání informací začínající u nižších organizačních struktur v rámci hierarchie daného podniku. Získané informace se posuzují z hlediska potřeb konkrétní organizace, vazeb na ostatní části systému, dostupnost zdrojů apod. Na tomto základě jsou poté specifikovány alternativy možných projektů. V následující fázi je proveden výběr projektů k realizaci (zde je možné použít rozhodovací analýzu). Alternativa s nejlepším hodnocením představuje nejvhodnější projekt k realizaci. První etapa pak musí být završena rozhodnutím, ohledně pokračování daného projektu. Zde je specifikováno,

zdali je možné přejít na další etapu životního cyklu, nebo existuje potřeba dodatečných změn, nebo je nutné postup na další etapu zamítnout. [KALUŽA, 2010]

2.7.2. Zahájení a plánování projektů

Jestliže bylo v rámci etapy identifikace a výběr projektů nejasně specifikováno obsahové vymezení projektů, v rámci této etapy musí být jednoznačně vymezeno. Toto je podmínkou realistického ohodnocení projektů.

V rámci zahájení projektů je ustanoven tým řešitelů. Tento tým by měl mít jasnou personální strukturu. Jádrem týmu řešitelů se následně začne zabývat plánováním. Paralelně s tím je založena systémová dokumentace, za níž je zodpovědný konkrétní člen týmu. Je též stanoven formát dokumentace a způsob přístupu členů týmu.

Fáze plánování projektů je specifikován pomocí činností, které vedou ke zpracování plánu projektu pro každý jednotlivý projekt řešící daný informační systém. Plán projektu má jasně danou specifickou strukturu. V rámci této fáze je také možné ustanovit informativní materiál, pro budoucí uživatele systému. Tento materiál je založen na plánu projektu a jeho smyslem je připravit uživatele na budoucí práci se systémem. [KALUŽA, 2010]

V této etapě je také nutné zabývat se proveditelností projektu. Ta je ovlivněna množstvím faktorů, které je nutné vzít do úvahy. Zde je řeč zejména o technických faktorech, které se týkají především schopnosti organizace implementovat daný informační systém. V rámci každého projektu je možné identifikovat specifická rizika jako například:

- nedosažení očekávaných přínosů,
- přečerpání stanovených nákladů,
- překročení časového plánu projektu,
- nedosažení požadované funkčnosti,

Částečný vliv na proveditelnost projektu mají také legislativní aspekty. Zde je důležité stanovit, do jaké míry daný systém podléhá platným legislativním normám. Důležitá je zde také míra podpory ústředních osob dané organizace, která je očekávaná pro realizaci řešeného projektu. Dále musí být jasně definována očekávaná efektivnost projektu a je stanovena rozdílem přínosů a nákladů. Zde kromě měřitelných, tzv. přímých přínosů (zvýšení prodeje) existují i přínosy, které je v praxi obtížné vyčíslit, to

jsou tzv. nepřímé přínosy (zlepšení jména firmy). Totéž platí také v případě nákladů a tudíž je v této souvislosti řeč o tzv. přímých (mzdové náklady) a nepřímých (ztráta dobré pověsti) nákladech. Vzhledem k tomu, že efektivnost je stanovována v delším časovém horizontu, zavádí se zde metody pro výpočet současné hodnoty (PV - Present Value), nebo návratnosti investic (ROI - Return on Investment). Tyto hodnoty pak hrají spolu s hodnocením technických a legislativních faktorů zásadní roli v rozhodování o proveditelnosti projektu. Na základě těchto rozhodnutí je poté případně vydáno doporučení k realizaci projektu. [KALUŽA, 2010]

V závěrečné fázi této etapy by měla být provedena prezentace projektů a jejich následné schválení. Hlavním důvodem k prezentaci daných projektů je informační potřeba budoucích uživatelů i vedení dané společnosti. Zde by měli být informováni o plánovaném průběhu daných projektů. Na základě vzniklých připomínek je možné celý proces upravit. Dále jsou pak předložené projekty schvalovány příslušnými orgány, či osobami.

2.7.3. Analýza současného stavu

Následující etapou je analýza současného stavu. Tato etapa spolu s následující tvoří základ struktury životního cyklu vývoje IS. Zde je specifikováno fungování dosavadního systému, zhodnotí se jeho nedostatky a navrhnou možná zlepšení. Tuto etapu lze rozdělit do tří fází:

- určení požadavků,
- modelování procesů, algoritmů a dat,
- výběr nejlepší alternativy

Během fáze určení požadavků je podstatné shromáždit informace o fungování současného systému a požadavky na systém nový. Ke sběru informací lze využít mnoho metod jako např. rozhovor, dotazník, analýzu písemností, pozorování apod. Každá z metod má své výhody i nevýhody a proto se často používá jejich kombinace. [KALUŽA, 2010]

Pro následující fázi je pak charakteristické modelování procesů, algoritmů a logické struktury dat. K těmto účelům se využívají tzv. CASE (Computer-aided Software Engineering) nástroje. Zde je možné modelovat jak systém stávající, tak systém budoucí, přičemž oba přístupy mají své zastánce. Pro modelování procesů je

možné využít například diagram hierarchické struktury, kontextový diagram, diagram datových toků, nebo diagram případů užití. Tyto diagramy mohou pomoci vzájemné interakci mezi uživatelem a systémovým analytikem. Pro modelování algoritmů je pak možné použít nástroje, jakými jsou např. diagram aktivit, sekvenční diagram, stavový diagram, nebo diagram přehledu interakcí. V rámci tvorby konceptuálního datového modelu je možné použít například E-R model, diagram tříd, nebo diagram objektů. [KALUŽA, 2010]

Tato etapa je pak uzavřena fází výběru nejlepší alternativy. Obvykle je formulováno více alternativ, ze kterých se vybírá ta nejvhodnější. Pravidlem může být formulování tří různých alternativ (konzervativní, progresivní a jejich kombinace), přičemž konzervativní alternativa je založena na zlepšení stávajícího řešení, zásadní změna systému je reprezentována progresivní alternativou a třetí alternativa je pak kompromisem mezi uvedenými. Výběr nejvhodnější alternativy může být založen na výsledcích provedené rozhodovací analýzy. Vybranou alternativu je možné ještě před zahájením následující etapy upravit a to v rámci plánu projektu, řešitelského týmu, časového plánu a dalších. [KALUŽA, 2010]

2.7.4. Návrh nového řešení

Na základě analýzy z předchozí etapy je v této etapě vytvořen detailní datový model, vstupní a výstupní komunikace, algoritmy a architektura programového řešení v rámci počítačových sítí. Tuto etapu lze opět rozdělit do několika fází:

- logické datové modelování,
- návrh formulářů a sestav,
- návrh dialogů,
- návrh systémové a programové struktury,
- návrh architektury distribuce systému.

Ve fázi logického datového modelování je vycházeno z vytvořeného konceptuálního datového modelu z předchozí etapy. Pro tyto účely je používán převážně relační přístup datového modelování, díky kterému je daný model aplikovatelný na relační (objektově-relační) databázi. Vytvořená soustava relací je reprezentována grafickou, nebo textovou formou. Grafická forma je sice přehlednější, ale může zde vzniknout problém se zobrazením relací s mnoha atributy. Dále je pro každý specifikovaný atribut zaveden datový typ a specifikovány přípustné a implicitní hodnoty. [KALUŽA, 2010]

Z pohledu uživatele je následující fáze tou nejvýznamnější v rámci této etapy. V této fázi jsou sestaveny návrhy vstupních formulářů a výstupních sestav. Tyto komponenty by měli umožňovat komfortní zadávání dat do systému na straně jedné a poskytovat informace na straně druhé. Formuláře i sestavy jsou modelovány na základě diagramů datových toků, vytvořených v předchozí etapě. Při návrhu formulářů a sestav je důležité dodržet určité zásady (rovnoměrné rozložení dokumentu, snadná navigace, jasný a výstižný text apod.).

Fáze návrhu dialogů se týká především specifikace interakce mezi uživatelem a informačním systémem. Zde existuje množství metod, jejichž použití umožňuje vzájemnou interakci. Mezi tyto metody patří například příkazový jazyk, formulář, nabídka, nebo objekt. Zatímco příkazový jazyk je již zastaralou, metodou ostatní zmiňované metody jsou běžně používány.

Ve fázi návrhu systémové a programové struktury je definován celkový popis systému (účel, platforma, omezení, předpoklady), popis uživatelského rozhraní, popis jednotlivých programů a jejich částí, popis bezpečnosti dat a zpracování. Dále jsou zde uvedeny specifikace výkonnostních parametrů, kvality softwaru a způsobu testování programů.

V poslední fázi této etapy je řešena stavba systému v rámci počítačových sítí. Podle rozmístění komponent (prezentace dat, aplikační logika, řízení dat) mezi klienty a servery lze rozlišit tyto styly kooperativního zpracování: distribuovaná prezentace, vzdálená prezentace, vzdálené řízení dat, distribuovaná aplikace, distribuované řízení dat, distribuované zpracování.[KALUŽA, 2010]

2.7.5. Zavedení nového řešení

Touto etapou jsou završeny práce vývojářů na novém informačním systému. Tuto etapu lze rozdělit do následujících fází:

- kódování,
- testování,
- instalace,
- finalizace dokumentace,
- školení a konzultace.

V rámci kódování jsou převáděny návrhy systému do formy programového kódu. K tomu se využívá programovacích jazyků a vývojových prostředí. Poté následuje fáze testování programů a systému. Zde je hlavním cílem identifikace a oprava chyb, které jsou obsaženy ve vytvořených programech. Jedná se o zásadní implementační činnost a na kvalitě testování následně závisí úspěšné provozování systému. Proces testování by měl být plánován, měla by být vytvořena testovací data, na nichž se následně testování provádí (alfa testování). Za účelem testování lze použít i data reálná (beta testování). Účelem této fáze je prověření funkčnosti nového systému, odzkoušení jeho reakcí a výkonnosti a prověření zabezpečení.[KALUŽA, 2010]

Další fází této etapy je instalace systému. Zde jde především o náhradu stávajícího systému novým. Je zde také prováděna konverze dat ze stávajícího řešení do tvaru odpovídajícímu novému systému. Instalaci je možné provést několika způsoby, kterými jsou: přímá instalace, paralelní instalace, lokální instalace, fázová instalace. Přímá instalace je volena, jestliže stávající systém je třeba v určitý čas plně nahradit systémem novým. Jedná se o nejvíce rizikový přístup k instalaci, především kvůli možnému výskytu chyb. Paralelní instalace je naopak velmi bezpečná, ale zároveň také nákladná. Smyslem této instalace je časově vymezený paralelní běh obou systémů. Lokální instalace je pak kompromisním řešením, které kombinuje oba předchozí přístupy. Nový systém se nainstaluje pouze v jedné části organizace, zatímco v ostatních částech je stále nasazen systém původní. Následuje postupné rozšiřování systému po celé organizaci. Fázová instalace je také formou kompromisního řešení. Zde se instaluje napříč podnikem pouze část nového systému, která je postupně obohacována o další programové části, až je nakonec sestaven celý systém.[KALUŽA, 2010]

Nedílnou částí této etapy je také fáze finalizace dokumentace. V této fázi je dokončen proces tvorby dokumentace, který probíhá ve všech etapách životního cyklu vývoje informačního systému. Z pohledu uživatele systému lze dokumentaci rozdělit na systémovou a uživatelskou. Systémová dokumentace je určena především vývojářům systému, zatímco uživatelská jeho uživatelům. Jako příklad uživatelské dokumentace může sloužit uživatelská příručka, nebo příručka pro správce systému. Po instalaci systému a dokončení dokumentace následuje fáze školení a konzultace uživatelů. Školení uživatelů je klíčové pro úspěch nového systému. Cílem je naučit uživatele obsluhovat systém a zároveň podpořit jeho kladný vztah k novému systému.

Konzultace pak probíhají po zavedení systému a zajišťují tak uživatelům podporu při vzniku specifických problémů. [KALUŽA, 2010]

2.7.6. Údržba systému

Údržba systému je poslední etapou životního cyklu vývoje IS. Tato etapa spočívá v údržbě zavedeného systému při jeho rutinním provozu. Existuje několik faktorů, které zapříčiňují nezbytnost údržby. Jedná se především o chyby, které mohou vznikat i po delším provozu systému a dále pak o změny tykající se požadavků uživatelů, legislativního prostředí apod. Je zde možné stanovit základní typy údržby:

- opravná,
- adaptační,
- zdokonalující,
- preventivní.

V rámci opravné údržby jsou odstraňovány chyby, které se objevily již v rámci vývoje systému. V rámci adaptační údržby jsou prováděny modifikace systému na základě uživatelských požadavků, změny legislativy, tržních změn apod. Zdokonalující údržba pak spočívá především ve zlepšování výkonu a využitelnosti systému. Modifikace systému, které se týkají předcházení budoucím problémům, se pak vykonávají v rámci preventivní údržby. Při provádění jakéhokoliv typu údržby může dojít ke vzniku dalších chyb, které zapříčiňují potřebu další údržby. [KALUŽA, 2010]

3. Analýza současného stavu a identifikace požadavků zadavatele

Ze specifikace životního cyklu vývoje informačních systémů, který byl uveden v předchozí kapitole, vyplývá 6 vývojových etap. Tato diplomová práce je zaměřena především na vytvoření nového rezervačního systému, přičemž samotné kódování je součástí páté etapy. Nicméně nejprve je zapotřebí zjistit požadavky zadavatele a následně systém navrhnout. Tyto činnosti jsou specifikovány ve třetí a čtvrté etapě životního cyklu. Těmito etapami jsou: analýza současného stavu a návrh nového řešení.

Tato kapitola je věnována analýze současného stavu, kde je specifikován současný systém rezervace lední plochy zimního stadionu. V rámci této kapitoly jsou také specifikovány požadavky zadavatele na tento systém. Vzhledem k absenci první a druhé vývojové etapy v této diplomové práci je zde také stručně popsán objekt zadavatele projektu.

3.1. Popis objektu zadavatele

Zimní stadion Litomyšl je ve správě společnosti Městské služby Litomyšl s.r.o. Městské služby Litomyšl s. r. o. byly založeny na základě usnesení zastupitelstva města Litomyšle ze dne 18. 2. 2010. Tímto usnesením bylo stanoveno město Litomyšl jediným společníkem této společnosti. Společnost MSL s.r.o. vznikla transformací původní příspěvkové organizace města, Technických služeb města Litomyšle. Činnost Městských služeb s.r.o. byla zahájena k 1. 7. 2010 přičemž byly zřízeny dvě její divize:

- divize technická,
- divize sportovních a rekreačních služeb.

Hlavní náplní společnosti je zajišťovat specifické práce pro město Litomyšl. Mezi tyto práce patří údržba místních komunikací, chodníků, mostů a lávek, údržba dopravního značení, dopravní mechanizační činnost, údržba zeleně, veřejného osvětlení a hřbitova. Dále pak evidence hrobů, evidence budov, agenda nájemného a služeb souvisejících s užíváním bytů a nebytových prostor, správa, provoz a údržba veškerých sportovišť včetně dětských hřišť a zajišťování kulturních, společenských a

sportovních akcí. Některé z těchto služeb jsou poskytovány i pro veřejnost (např. údržba zeleně, doprava).

Zimní stadion Litomyšl tedy spadá pod technickou divizi MSL s.r.o. Tento stadion byl vybudován roku 1977, přičemž v roce 2002 byla vybudována nová tribuna pro 1600 stojících diváků. V roce 2004 prošel zimní stadion výraznou modernizací, kdy byl, podle projektu Ing. Arch. Aleše Buriana, zastřešen a následně roku 2005 bylo dokončeno opláštění východní a jižní strany, kde byla vybudována i nová tribuna pro 220 diváků (160 sedících, 60 stojících). V roce 2006 pokračovala modernizace opláštěním severní a západní strany, současně byla nainstalována nová výsledková tabule a rekonstruováno chlazení lední plochy. Stadion byl poté také vybaven plochou na inline hokej a je tedy možné jeho celoroční využití.

3.2. Analýza současného stavu

Specifikace současného stavu probíhala na základě rozhovorů s jednatelem společnosti Městské služby Litomyšl s.r.o. a správcem zimního stadionu, jakožto zaměstnancem této společnosti. Z provedených rozhovorů vyplývá níže uvedený popis užívaného procesu rezervace hrací plochy zimního stadionu.

Hrací plocha je pronajímána veřejnosti a sportovním organizacím po 1,5 hodinových intervalech, kterým je přiřazena konkrétní cena. Rezervovat hrací plochu je možné v časovém intervalu od 7 do 22 hodin. Každý den je tak rozdělen do 10 rezervovatelných jednotek. Některé tyto jednotky jsou ovšem vyhrazeny sportovním klubům a tedy veřejnost nemá možnost v těchto termínech hrací plochu využívat. Současná evidence rezervací funguje na bázi tabulek (vytvořených nástrojem Excel ze sady Microsoft Office), do kterých správce sportovního areálu vepisuje obsazené jednotky. Případné rezervace probíhají tak, že v případě požadavku od klienta správce porovná požadovaný termín s danou tabulkou a pokud je odpovídající jednotka neobsazená zarezervuje ji. V praxi pak taková situace vypadá tak, že zájemce o pronájem hrací plochy musí buď telefonicky, nebo osobně kontaktovat správce zimního stadionu (nebo jím pověřenou osobu) s požadavkem na rezervaci konkrétní jednotky. Správce náhledem do tabulky zjistí, zdali je požadovaná jednotka volná a případně zapíše jméno objednavatele. Problém nastává v případě, že požadovaná

jednotka je již obsazená. V tomto případě musí zájemce zvolit jinou jednotku a znovu se zeptat na její dostupnost (pokud má o jinou jednotku zájem).

Tento systém je z několika důvodů nevyhovující. Z pohledu zájemce o hrací plochu je problémem to, že nemá k dispozici příslušnou tabulku termínů a tudíž nemá možnost zjistit dostupnost konkrétních jednotek. Také způsob rezervace je značně nekomfortní, protože danou rezervaci nelze vyřídit jinak, než telefonicky, nebo osobně. Z pohledu provozovatele je hlavním problémem vytížení pracovní síly v podobě zaměstnance, který musí být celou pracovní dobu přítomný v kanceláři a manuálně vyřizovat všechny rezervace. Dále zde chybí jakákoliv evidence klientů, což může způsobovat problémy v podobě nespolehlivých zákazníků.

3.2.1. Požadavky zadavatele

V rámci analýzy současného stavu je také důležité správně specifikovat požadavky zadavatele na budoucí systém. Velmi důležitou roli zde hraje kvalita komunikace mezi zadavatelem projektu a jeho realizátorem. Součástí této komunikace by měli být také budoucí uživatelé systému. Pro shromáždění potřebných informací byla opět použita metoda rozhovoru, která probíhala ve dvou rovinách. Jednak byly specifikovány požadavky na vzhled a jednak požadavky na funkci budoucího systému. Zmíněné rozhovory probíhaly opět s jednatelem společnosti MSL s.r.o. a správcem zimního stadionu. Výsledkem těchto rozhovorů jsou následující specifikace:

- požadavky na vzhled
 - jednoduchý a čistý design,
 - použití responzivního web designu,
 - intuitivní rozmístění ovládacích prvků,
 - umístění fotografií zimního stadionu,
 - umístění mapy se zobrazením ZS a jeho adresou,
 - prezentace obsazenosti jednotlivých termínů formou denních tabulek,
- požadavky na funkci
 - dostupnost prostřednictvím webové stránky,
 - registrace uživatelů,
 - přihlašování uživatelů,
 - odhlašování uživatelů,
 - zobrazování obsazených a volných termínů všem uživatelům,

- možnost rezervace termínů pouze přihlášeným uživatelům,
- možnost správy vlastních rezervací pro uživatele,
- možnost editace uživatelských údajů,
- správa uživatelů dostupná pouze správci,
- správa rezervací dostupná pouze správci,
- vytváření měsíčních pohledů.

Na základě zjištěných zadavatelských požadavků na funkci byla sestavena tabulka uživatelských požadavků (Tab. 3.1). Na základě této tabulky byl pak ke každému uživatelskému požadavku vytvořen případ užití. Prostřednictvím případů užití je získán první pohled na rozsah systému. Jednotlivé případy užití jsou zachyceny tabulkami 3.2 – 3.11. Na závěr této kapitoly byl vytvořen diagram aktivit, který zachycuje chování uživatele v budoucím systému. Tento diagram byl pak sestaven na základě uvedených případů užití. Pomocí tohoto diagramu jsou modelovány jednotlivé případy užití jako posloupnost akcí. Tento grafický pohled by měl sloužit především pro větší přehlednost a lepší představu o rozsahu daného systému. Sestaveny byly dva diagramy aktivit, jeden pro modelování chování uživatele a druhý (velmi podobný) pro modelování chování správce. Zde je uveden první zmiňovaný (Obr. 3.1)

Id	Uživatelský požadavek
P1	Registrovat se do rezervačního systému
P2	Přihlásit se do rezervačního systému
P3	Odhlásit se z rezervačního systému
P4	Editovat uživatelské údaje
P5	Zobrazit rozpis termínů
P6	Rezervovat volný termín
P7	Spravovat své rezervace
P8	Spravovat všechny rezervace
P9	Spravovat uživatele
P10	Vytvářet měsíční pohled

Tab. 3.1 Uživatelské požadavky

Případ užití: Registrace do rezervačního systému
ID: UC1
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel má přístup k internetu 2. Systém správně funguje a je dostupný
Tok událostí: 1. Uživatel má zájem o registraci do systému 2. Uživatel zadá registrační údaje 3. Uživatel potvrdí registraci 4. Systém ověří správnost vyplněných údajů 5. KDYŽ (údaje správně vyplněny) 5.1 Uživatel je uložen do databáze uživatelů 6. JINAK 6.1 Uživatel je vrácen na registrační obrazovku
Následné podmínky: Uživatel zaregistrován

Tab. 3.2 Případ užití - registrace do rezervačního systému

Případ užití: Přihlášení do rezervačního systému
ID: UC2
Účastníci: Uživatel Správce
Vstupní podmínky: 1. Uživatel (správce) má přístup k internetu 2. Systém správně funguje a je dostupný 3. Uživatel (správce) je registrován v systému
Tok událostí: 1. Uživatel (správce) má zájem přihlášení do systému 2. Uživatel (správce) zadá přihlašovací údaje 3. Systém ověří správnost přihlašovacích údajů 4. KDYŽ (přihlašovací údaje v pořádku) 4.1 Uživatel přihlášen 5. JINAK 5.1 Uživatel (správce) vrácen na přihlašovací obrazovku
Následné podmínky: Uživatel (správce) přihlášen

Tab. 3.3 Případ užití – přihlášení do rezervačního systému

Případ užití: Odhlášení z rezervačního systému
ID: UC3
Účastníci: Uživatel Správce
Vstupní podmínky: 1. Uživatel (správce) má přístup k internetu 2. Systém správně funguje a je dostupný 3. Uživatel (správce) je přihlášen v systému
Tok událostí: 1. Uživatel (správce) ukončil práci v systému 2. Uživatel (správce) má zájem o odhlášení ze systému 3. Uživatel (správce) se odhlašuje ze systému
Následné podmínky: Uživatel (správce) odhlášen

Tab. 3.4 Případ užití – Odhlášení z rezervačního systému

Případ užití: Editace uživatelských údajů
ID: UC4
Účastníci: Uživatel Správce
Vstupní podmínky: 1. Uživatel (správce) má přístup k internetu 2. Systém správně funguje a je dostupný 3. Uživatel (správce) je přihlášen v systému
Tok událostí: 1. Uživatel (správce) má zájem o změnu uživatelských údajů 2. Uživatel (správce) zobrazí uživatelské údaje 3. Uživatel (správce) změní uživatelské údaje 4. Systém ověří správnost vyplněných údajů 5. KDYŽ (údaje správně vyplněny) 5.1 Změny jsou uloženy do databáze uživatelů 6. JINAK 6.1 Změny nejsou provedeny a uživatel je vrácen na přehled uživatelských údajů
Následné podmínky: Uživatelské údaje změněny

Tab. 3.5 Případ užití – Editace uživatelských údajů

Případ užití: Zobrazení rozpisu termínů
ID: UC5
Účastníci: Uživatel Správce
Vstupní podmínky: 1. Uživatel (správce) má přístup k internetu 2. Systém správně funguje a je dostupný
Tok událostí: 1. Uživatel (správce) má zájem o zobrazení rozpisu termínů 2. Uživatel (správce) zobrazí rozpis termínů
Následné podmínky: Rozpis termínů je zobrazen

Tab. 3.6 Případ užití – Zobrazení rozpisu termínů

Případ užití: Výběr a rezervace termínu
ID: UC6
Účastníci: Uživatel Správce
Vstupní podmínky: 1. Uživatel (správce) má přístup k internetu 2. Systém správně funguje a je dostupný 3. Uživatel (správce) je přihlášen v systému
Tok událostí: 1. Uživatel (správce) má zájem o rezervaci termínu 2. Uživatel (správce) zobrazí přehled termínů 3. Uživatel (správce) vybere volný termín 4. Uživatel (správce) zarezervuje termín 5. Termín je uložen do databáze jako zarezervovaný daným uživatelem (správcem)
Následné podmínky: Termín zarezervován

Tab. 3.7 Případ užití – Výběr a rezervace termínu

Případ užití: Správa uživatelských rezervací
ID: UC7
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel má přístup k internetu 2. Systém správně funguje a je dostupný 3. Uživatel je přihlášen v systému
Tok událostí: 1. Uživatel má zájem o zobrazení a případné zrušení existující rezervace 2. Uživatel zobrazí přehled uživatelských rezervací 3. Uživatel vybere rezervaci, kterou chce zrušit 4. Systém ověří, zda je možné rezervaci zrušit 5. KDYŽ (rezervaci je možné zrušit) 5.1 Uživatel potvrdí zrušení rezervace 5.2 Rezervace je zrušena 6. JINAK 6.1 Rezervace není zrušena
Následné podmínky: Rezervace zrušena / nezrušena

Tab. 3.8 Případ užití – Správa uživatelských rezervací

Případ užití: Správa všech rezervací
ID: UC8
Účastníci: Správce
Vstupní podmínky: 1. Uživatel má přístup k internetu 2. Systém správně funguje a je dostupný 3. Správce je přihlášen v systému
Tok událostí: 1. Správce má zájem o zobrazení a případnou editaci existující rezervace 2. Správce zobrazí přehled rezervací 3. Správce vybere požadovanou rezervaci 4. Správce edituje vybranou rezervaci 5. Změna je uložena do databáze
Následné podmínky: Rezervace editována

Tab. 3.9 Případ užití – Správa všech rezervací

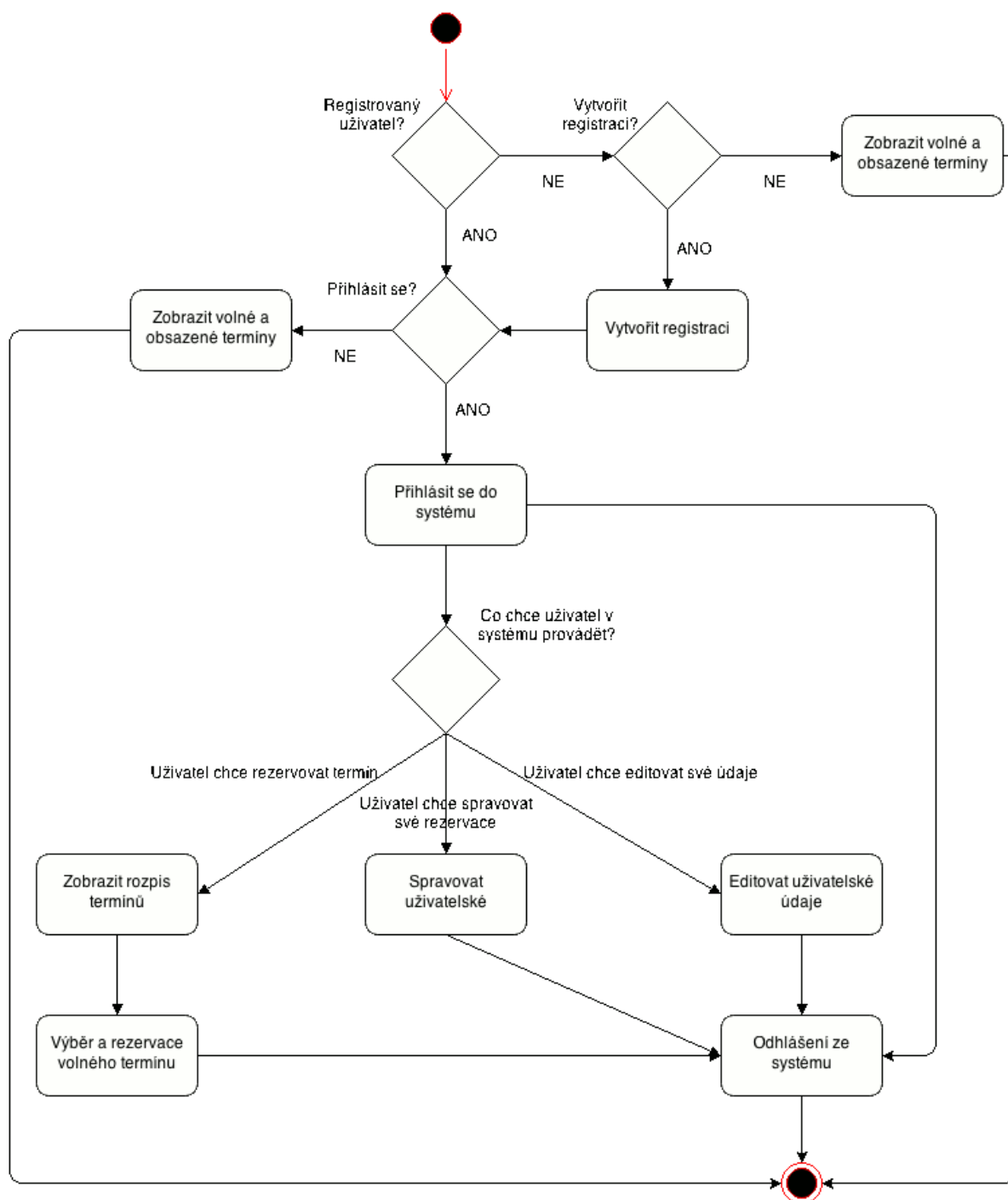
Případ užití: Správa všech uživatelů
ID: UC9
Účastníci: Správce
Vstupní podmínky: 1. Uživatel má přístup k internetu 2. Systém správně funguje a je dostupný 3. Správce je přihlášen v systému
Tok událostí: 1. Správce má zájem o zobrazení a případnou editaci registrovaných uživatelů 2. Správce zobrazí přehled uživatelů 3. Správce vybere požadovaného uživatele 4. Správce edituje vybraného uživatele 5. Změna je uložena do databáze
Následné podmínky: Uživatel editován

Tab. 3.10 Případ užití – Správa všech uživatelů

Případ užití: Vytváření měsíčního pohledu
ID: UC10
Účastníci: Správce
Vstupní podmínky: 1. Uživatel má přístup k internetu 2. Systém správně funguje a je dostupný 3. Správce je přihlášen v systému
Tok událostí: 1. Správce má zájem o vytvoření měsíčního pohledu 2. Správce zobrazí přehled rezervací 3. Správce zvolí položku měsíční pohled 4. Systém vygeneruje a zobrazí měsíční pohled
Následné podmínky: Měsíční pohled vytvořen

Tab. 3.11 Případ užití – Vytváření měsíčního pohledu

Sestavení případů užití dopomáhá ke komplexnějšímu pohledu na plánovaný systém a jeho rozsah. Díky uvedeným případům užití je tak možné pozorovat řešení jednotlivých uživatelských požadavků v rámci systému, respektive chování uživatele při řešení těchto požadavků. Například prostřednictvím případu užití UC1 je zachyceno chování uživatele v systému při vyřizování uživatelského požadavku P1, který reprezentuje potřebu uživatele registrovat se do systému. Jak je z tabulky 3.2 patrné, musejí být nejprve splněny vstupní podmínky. To znamená, že v případě nedostupnosti připojení k internetu, nebo nedostupnosti daného systému, není možné uživatele zaregistrovat. Pokud jsou tyto podmínky splněny a uživatel má stále zájem o registraci v systému, může uživatel zahájit práci se systémem. V tomto případě tak učiní zadáním požadovaných registračních údajů a jejich následným odesláním. V rámci systému jsou následně vyplněné údaje ověřeny. Pokud byly správně zadány, uživatel je uložen do databáze uživatelů. V opačném případě je uživatel přesměrován zpět na registrační obrazovku. Aby došlo ke splnění uvedeného uživatelského požadavku, musí být uživatel do systému zaregistrován. Tento fakt je tedy uveden jako následná podmínka. Obdobným způsobem je možné interpretovat jakýkoliv z dalších uvedených případů užití.



Obr. 3.1 Diagram aktivit z pohledu chování uživatele

4. Návrh řešení a implementace rezervačního systému

V této kapitole bude věnován prostor především návrhu nového systému a následné jeho implementaci. Tyto kroky jsou specifikovány ve čtvrté a páté etapě životního cyklu vývoje informačního systému. V rámci návrhu nového řešení bude sestaven jednoduchý relační datový model, poté bude zhotoven grafický návrh rozmístění jednotlivých komponent v rámci webové stránky a celkový návrh vzhledu systému. Dále budou navrženy vstupní formuláře a výstupní sestavy. Nakonec zde budou popsány jednotlivé komponenty systému. V rámci implementace nového rezervačního systému bude provedeno jeho samotné kódování.

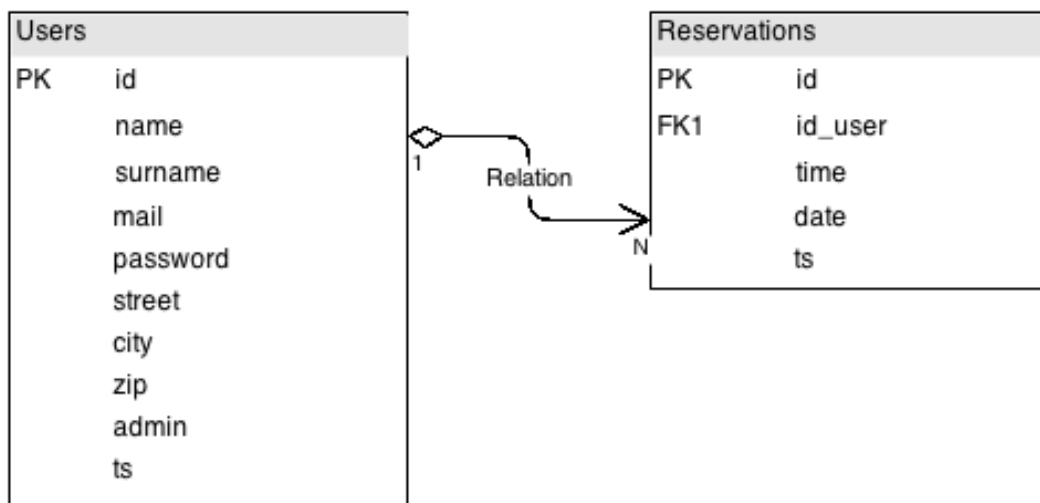
4.1. Návrh řešení

Jak již bylo řečeno, nejprve zde bude sestaven relační datový model. Výsledkem tohoto modelu je grafické i textové zobrazení relací a jejich atributů. Těmto atributům je následně přidělen datový typ a dále jsou implicitní hodnoty. Nejprve je model specifikován textovou (indexem PK je označen primární klíč, indexem FK je označen cizí klíč):

Users (id_{PK} name, surname, mail, password, street, city, zip, admin, ts),

Reservations (id_{PK}, id_user_{FK}, time, date, ts).

Jak je patrné již z textového zobrazení, je v rámci databáze navrhovaného systému počítáno se dvěma databázovými tabulkami. Do tabulky Users budou ukládána data týkající se uživatelů systému a do tabulky Reservations budou ukládána data týkající se samotných rezervací. Z důvodu zpřehlednění této situace je tento model reprezentován také grafickou formou (Obr. 4.1) Po stanovení tabulek a příslušných atributů je nutné určit datový typ daných atributů. Jak je patrné z tabulek 4.1 a 4.2, textovým hodnotám je standardně přidělen datový typ varchar (variabilní délka) a číselným hodnotám int (4 byty), mediumint (3 byty), nebo tinyint (1 byte). Dále jsou zde použity datové typy date, time a timestamp. Jak již jejich název napovídá, jedná se o datové typy použité pro uložení data, času a časového razítka. Co se týče implicitních hodnot, ty jsou nastaveny u obou tabulek pouze u atributů ts, kam se ukládá hodnota časového razítka. Oběma atributům s názvem id je dále přidělena vlastnost AUTO INCREMENT, která zabezpečí automatickou inkrementaci daného čísla.



Obr. 4.1 relační datový model

Název	Typ
id	int
name	varchar(100)
surname	varchar(100)
mail	varchar(100)
password	varchar(50)
street	varchar(100)
city	varchar(100)
zip	mediumint(5)
admin	tinyint(1)
ts	timestamp

Tab. 4.1 Tabulka Users

Název	Typ
id	int
id_user	int
time	time
date	date
ts	timestamp

Tab 4.2 Tabulka Reservations

Poté co byl navržen relační datový model, bylo přistoupeno k návrhu vzhledu systému. Vzhledem k tomu, že budoucí rezervační systém bude prezentován formou webových stránek, týká se návrh vzhledu především rozložení webové stránky v rámci internetového prohlížeče a rozmístění jednotlivých komponent. Dále je pak sestaven grafický návrh vzhledu rezervačního systému, podle kterého bude posléze vytvořen design celého systému. Rozložení webové stránky a jednotlivých komponent bylo navrženo na základě konzultací se zadavatelem systému a jednotlivých

zadavatelských požadavcích. Na obrázku 4.2 je zobrazen návrh rozložení webové stránky pro hlavní stránku.

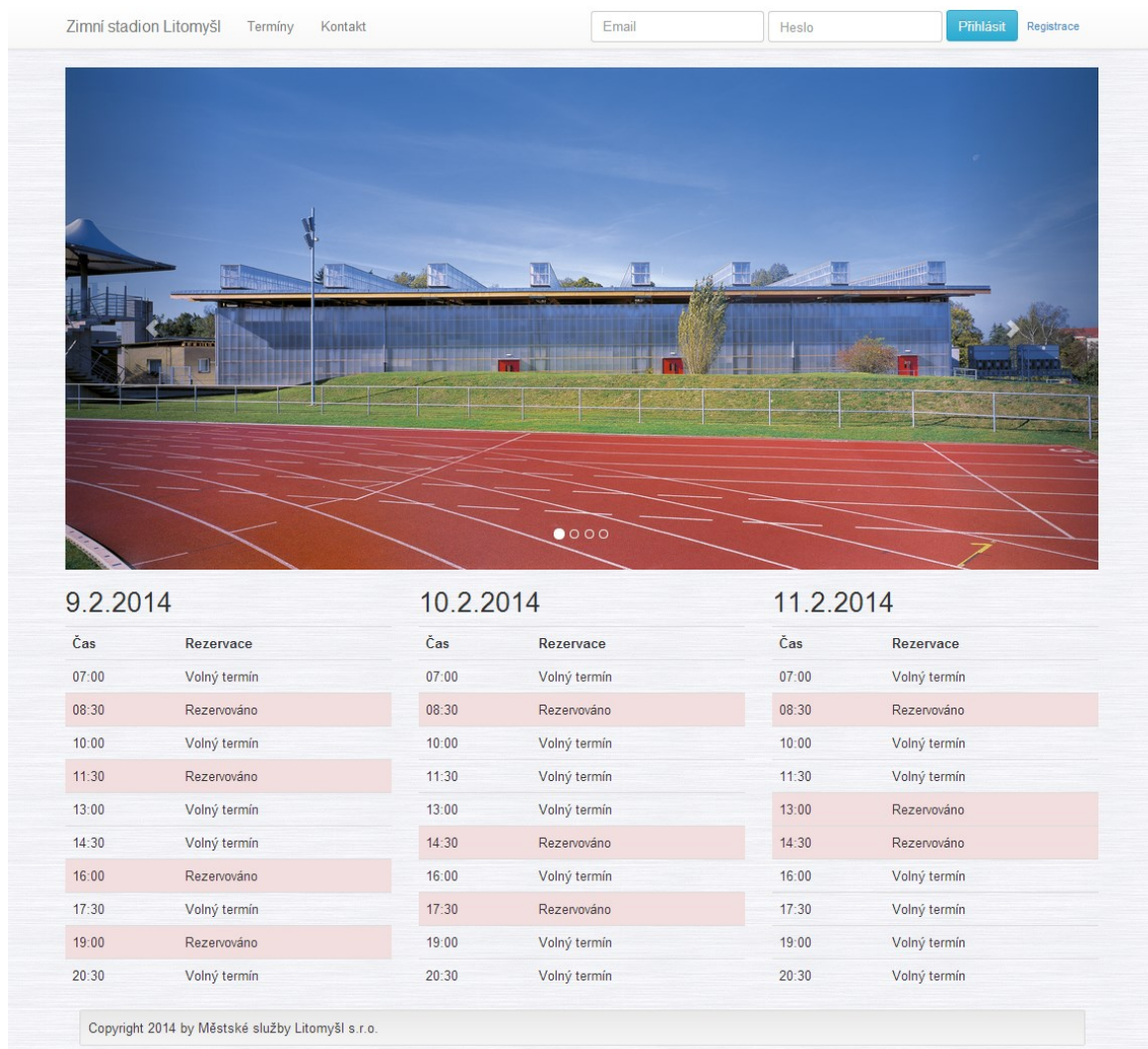
Název objektu	Navigace	Přihlašovací formulář
Fotogalerie		
Tabulka termínů	Tabulka termínů	Tabulka termínů
Patička		

Obr 4.2 Návrh rozložení webové stránky

Jak je z tohoto obrázku patrné, byl zvolen jednoduchý jednosloupcový rozvrh (layout) stránky. To by mělo zabezpečit přehlednost prezentované webové stránky a intuitivní ovládání celého systému. Do levého horního rohu je standardně umístěn název objektu (Zimní stadion Litomyšl), do hlavičky stránky je dále umístěna navigace a přihlašovací formulář. Dle požadavků zadavatele je zde také umístěna fotogalerie. Dále jsou zde umístěny tři tabulky pro zobrazení termínů. Tyto tabulky budou reprezentovat obsazenost hrací plochy pro aktuální dny. Z toho vyplývá, že rezervace pro nejbližší 3 dny (dnes, zítra, pozítří) může být provedena přímo z úvodní obrazovky, aniž by uživatel musel přejít na podrobnější výpis termínů. Na spodní straně stránky je umístěna patička, kde je možnost zobrazit například standardní informace o autorských právech apod.

Poté co byl vytvořen a zadavatelem odsouhlasen konečný návrh rozložení webové stránky, bylo přistoupeno k vytvoření konkrétního grafického návrhu vzhledu celého systému. Výsledný návrh vzhledu hlavní stránky (pro nepřihlášeného uživatele)

je zobrazen na obrázku 4.3. Tento návrh plně odpovídá předchozímu odsouhlasenému návrhu rozmístění komponent. Výsledný grafický návrh byl sestaven pomocí komponent nástroje Bootstrap, což je tzv. front-end framework pro vývoj webových projektů. Dále bude tento nástroj blíže specifikován. Do dříve určené oblasti webové stránky je umístěna fotogalerie. Zde byla zvolena forma zobrazení pouze jediné fotografie. Mezi jednotlivými fotografiemi bude možné přepínat pomocí navigace, nebo navigačních tlačítek umístěných na pravém a levém okraji aktuálně zobrazené fotografie. Jak je patrné z obrázku 4.3, v rámci budoucího systému bude umožněno nepřihlášeným uživatelům prohlížet obsazenost jednotlivých termínů, nebude však možná jejich rezervace. Zobrazení obsazenosti jednotlivých termínů je prezentováno zde prezentováno formou přehledných denních tabulek. Toto je opět plně v souladu se zadavatelskými požadavky. Po definitivním dokončení návrhu vzhledu celého systému byl tento výsledný návrh předán ke schválení zadavateli.



Obr. 4.3 Návrh vzhledu hlavní strany systému

Po odsouhlasení výsledného grafického návrhu vzhledu systému bylo přikročeno k další fázi návrhu. V této fázi jsou modelovány vstupní formuláře a výstupní sestavy. V případě tohoto rezervačního systému se jedná především o přihlašovací formulář, registrační formulář a formulář pro zadávání data. Co se týče sestav, jedná se především o sestavy generované pro potřebu správce systému, jakými jsou např. seznam rezervací a měsíční pohled na obsazenost termínů a dále sestavy pro uživatele systému. Přihlašovací formulář slouží pro přihlášení uživatele do systému. Zde uživatel zadá registrovanou e-mailovou adresu a registrované heslo a následně zadané údaje odešle ke zpracování. Při zadání řádných hodnot je pak uživatel přihlášen do systému. V opačném případě je uživateli zobrazeno hlášení o zadání nesprávných údajů. Grafický návrh tohoto formuláře je patrný již z obrázku 4.3, kde je přihlašovací formulář zobrazen v pravém horním rohu. Dále byl navržen registrační formulář, který slouží k registraci nového uživatele do rezervačního systému. Grafická podoba tohoto formuláře je znázorněna obrázkem 4.4. V případě tohoto vstupního formuláře budou zadané informace vkládány do databázové tabulky. Při návrhu je tedy nutné s tímto faktem počítat a navrhnout omezení pro zadávané hodnoty. Všechny vstupní pole bude tedy nutné vyplnit a bude omezen rozsah zadaných znaků. Dále pak bude nastaveno omezení na poli „PSČ“, kde bude možno zadat pouze číselnou hodnotu v přesné délce pěti znaků. Na poli „Email“ bude nastaveno omezení v podobě nutnosti zadání validní e-mailové adresy. Pole „Heslo“ bude nutné vyplnit minimálně 6 znaky. Do pole „Potvrzení hesla“ pak bude nutné zadat stejný řetězec jako do pole „Heslo“. Tento vstupní formulář je zde navržen tak, že v případě nedodržení těchto podmínek, nebude vůbec možné jeho odeslání ke zpracování.

Jméno	Příjmení
<input type="text" value="Jméno"/>	<input type="text" value="Příjmení"/>
Ulice	Město
<input type="text" value="Ulice"/>	<input type="text" value="Město"/>
PSČ	Email
<input type="text" value="PSČ"/>	<input type="text" value="mail"/>
Heslo	Potvrzení hesla
<input type="text" value="Heslo"/>	<input type="text" value="Heslo"/>
Minimálně 6 znaků	Zadejte stejné heslo
<input type="button" value="Registrovat se"/>	

Obr. 4.4 Návrh registračního formuláře

Formulář pro zadávání data je v tomto případě navržen jako rozevírací kalendář, ve kterém je možné vybrat požadované datum a to následně odeslat ke zpracování. Takovéto zobrazení vstupního formuláře usnadní uživateli orientaci v jednotlivých datech a zjednoduší tak interakci mezi uživatelem a systémem. Zde je velmi důležité navrhnout tento kalendář tak, aby uživatel nemohl zvolit termín v minulosti. Ve zmiňovaném kalendáři jsou pak uvedeny zkratky názvů jednotlivých dnů, název měsíce a kalendářní rok. Dále je zde zvýrazněno aktuální datum a umístěno tlačítko, na toto datum odkazující. Návrh tohoto formuláře je možné vidět na obrázku 4.5.

The screenshot displays a web application for a winter stadium reservation system. The header includes the stadium name 'Zimní stadion Litomyšl' and navigation links: 'Termíny', 'Moje rezervace', 'Správa rezervací', and 'Správa uživatelů'. A user profile 'Aleš Kalousek' and a 'Logout' button are also present. The main content area shows a welcome message and a date selection interface. A calendar for April 2014 is open, with the 23rd highlighted. Below the calendar, there are two columns of reservation slots for 23.04.2014 and 24.04.2014. Each slot includes a time (e.g., 07:00, 08:30) and a button to 'Rezervovat' (Reserve). Some slots are already reserved, indicated by a 'Rezervováno' status.

Po navržení vstupních formulářů bylo přikročeno k návrhu výstupních sestav. Pomocí těchto sestav jsou uživateli prezentovány požadované informace. Z tohoto pohledu jsou výstupní sestavy pro uživatele tou nejdůležitější součástí celého systému. V případě tohoto rezervačního systému jsou výstupní sestavy reprezentovány především jednotlivými denními tabulkami obsazenosti termínů (Obr. 4.5), tabulkami zobrazujícími jednotlivé uživatelské rezervace a tabulkami sestavovanými pro potřeby správce. Pomocí denních tabulek by měly být zobrazovány volné a obsazené časové jednotky. V případě obsazení dané časové jednotky by mělo být rozlišeno, zdali se jedná o rezervaci vytvořenou přihlášeným, nebo jiným uživatelem. V rámci sestav uživatelských rezervací bude možné zobrazit veškeré rezervace přihlášeného uživatele (s budoucím datem rezervace), přičemž zde bude

možné jednotlivé rezervace také rušit (pouze do 24 hodin před termínem dané rezervace). V případě sestav pro potřeby správce se jedná konkrétně o tabulky s přehledem všech uživatelů a jejich údajů a tabulky všech rezervací. Po konzultaci se správcem zimního stadionu byly nakonec tabulky všech rezervací navrženy tak, že bude možné zobrazit veškeré budoucí rezervace a rezervace, které proběhly za poslední měsíc. V rámci těchto sestav pak bude možné zobrazit datum a čas rezervace a jméno a příjmení uživatele, který danou rezervaci vytvořil. Dále zde bude možnost danou rezervaci přímo zrušit. Dle požadavků zadavatele bude také možné sestavit měsíční pohled na obsazenost jednotlivých termínů. V tomto případě pak bude možné zobrazit předchozí, aktuální a příští měsíc. Návrh sestavy měsíčního pohledu je zachycen na obrázku 4.6. Po sestavení návrhů vstupních formulářů a výstupních sestav byly tyto předány zadavateli k odsouhlasení.

Poté co byly zhotoveny a odsouhlaseny uvedené návrhy, bylo přistoupeno ke specifikaci budoucího systému a jeho komponent a to z hlediska použitých technologií. Vzhledem k tomu, že budoucí systém bude provozován ve webovém prostředí, bude jako základní technologie použit jazyk HTML verze 5. Pro definování vzhledu systému budou použity kaskádové styly (zde ve verzi CSS 3). Dále zde bude použit zmíněný framework Bootstrap. Tento framework byl vyvinut v prostředí společnosti Twitter a dále byl uvolněn pod licencí MIT, což umožňuje jeho volné použití pro soukromé i komerční účely.[Bootstrap] V podstatě se jedná o sadu nástrojů a komponent, které je možné použít pro definování uživatelského rozhraní. Tento nástroj zde bude použit k efektivnímu dosažení moderního, responzivního vzhledu. Pro vytváření skriptů, které budou prováděny na straně serveru, bude použit jazyk PHP (verze 5.5). Skripty napsané v tomto jazyce budou poskytovat nosnou funkcionalitu budoucího rezervačního systému. Pro definování skriptů prováděných na straně klienta pak bude použit jazyk JavaScript potažmo knihovna jQuery. Pro správu relační databáze bude použit databázový systém MySQL. Základními komponentami budoucího rezervačního systému tedy budou: relační databáze, serverové skripty, klientské skripty, HTML dokumenty a CSS dokumenty.

Měsíční pohled

Datum	7:00	8:30	10:00	11:30	13:00	14:30	16:00	17:30	19:00	20:30
01.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
02.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
03.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
04.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
05.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
06.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
07.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
08.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
09.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
10.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
11.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
12.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
13.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
14.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
15.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
16.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
17.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
18.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Rezervováno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno
19.04.2014	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Rezervováno	Neobsazeno	Neobsazeno
20.04.2014	Neobsazeno	Rezervováno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno
21.04.2014	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
22.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno
23.04.2014	Neobsazeno	Rezervováno	Neobsazeno	Rezervováno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
24.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
25.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno
26.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
27.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
28.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno
29.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno
30.04.2014	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Neobsazeno	Rezervováno	Neobsazeno	Neobsazeno	Neobsazeno

Copyright 2014 by Městské služby Litomyšl s.r.o.

Obr. 4.6 Návrh sestavy - měsíční pohled

4.2. Implementace řešení

Implementace nového řešení tvoří samostatnou etapu životního cyklu vývoje informačního systému. V této etapě je vytvořen samotný zdrojový kód jednotlivých komponent tvořeného rezervačního systému. Veškeré komponenty budou vytvořeny na základě dříve provedených analýz a návrhů. V této kapitole budou nastíněny

základní postupy a uvedeny ukázky ze zdrojového kódu jednotlivých komponent. Kompletní vytvořené soubory a databáze jsou pak obsaženy v příloze č. 1.

Při zahájení prací na implementaci nového rezervačního systému pro zimní stadion byly nejprve vytvořeny jednotlivé HTML dokumenty, jejichž vzhled byl dále definován pomocí kaskádových stylů. Vzhledem k tomu, že tyto soubory byly vytvořeny již v rámci vytváření návrhů vzhledu systému, byly v této fázi pouze finálně upraveny. Jak již bylo uvedeno, pro definování uživatelského rozhraní byly použity komponenty nástroje Bootstrap, které byly upraveny pro potřeby tvořeného rezervačního systému. Díky použití těchto komponent je velmi efektivním způsobem zajištěna responzivnost jednotlivých webových stránek tvořeného systému. Díky této vlastnosti by neměl vzniknout problém se zobrazením jednotlivých stránek napříč různými zařízeními. Toto bylo jedním z požadavků zadavatele. Celkem bylo vytvořeno 22 různých HTML dokumentů, které budou reprezentovat uživatelské rozhraní tvořeného systému. Způsob jejich zobrazení je pak definován prostřednictvím připojených CSS souborů. Zde je uveden zdrojový kód hlavní stránky, která je také vstupní stránkou celého systému. Návrh této stránky je také zobrazen na obrázku 4.3. Tento zdrojový kód již obsahuje některé PHP prvky, které byly doplněny v následujících krocích.

```
<!DOCTYPE html>
<html lang="cs">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Rezervační systém zimního stadionu
Litomyšl">
    <title>Zimní stadion Litomyšl
    </title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/bootstrap-theme.min.css" rel="stylesheet">
    <link href="css/styles.css" rel="stylesheet">
    <script src="js/jquery.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </head>
  <body>
    <?php
      session_start();
      require_once 'php/database.php';
      require_once 'php/tablegen.php';
    ?>
    <div class="navbar navbar-default" role="navigation">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
            <span class="sr-only">
            </span>
```

```

        <span class="icon-bar">
        </span>
        <span class="icon-bar">
        </span>
        <span class="icon-bar">
        </span>
    </button>
    <a class="navbar-brand" href="index.php">Zimní stadion
Litomyšl</a>
</div>
<div class="navbar-collapse collapse">
<?php
    if (isset($_SESSION["id"]) && $_SESSION["admin"] != 0 )
    include ("txt/menu_admin.txt");
    else if (isset($_SESSION["id"]) && $_SESSION["admin"] == 0)
    include ("txt/menu_user.txt");
    else
    include ("txt/menu_visiter.txt");
?>
</div>
</div>
</div>
<div class="container theme-showcase" role="main">
    <div class="col-8">
        <div id="carousel-example-generic" class="carousel slide" data-
ride="carousel">
            <ol class="carousel-indicators">
                <li data-target="#carousel-example-generic" data-slide-to="0"
class="active">
                </li>
                <li data-target="#carousel-example-generic" data-slide-to="1">
                </li>
                <li data-target="#carousel-example-generic" data-slide-to="2">
                </li>
                <li data-target="#carousel-example-generic" data-slide-to="3">
                </li>
            </ol>
            <div class="carousel-inner">
                <div class="item active">
                    
                    <div class="carousel-caption">
                    </div>
                </div>
            </div>
            <a class="left carousel-control" href="#carousel-example-generic"
data-slide="prev">
                <span class="glyphicon glyphicon-chevron-left">
                </span></a>
            <a class="right carousel-control" href="#carousel-example-generic"
data-slide="next">
                <span class="glyphicon glyphicon-chevron-right">
                </span></a>
            </div>
        </div>
    <div class="row">
        <div class="col-md-4 col-sm-6 col-lg-4"><h2>
            <?php echo date("d.m.Y"); ?></h2>
            <div class="table-responsive">
                <table class="table table-hover">
                    <th>Čas</th>
                    <th>Rezervace</th>

```

```

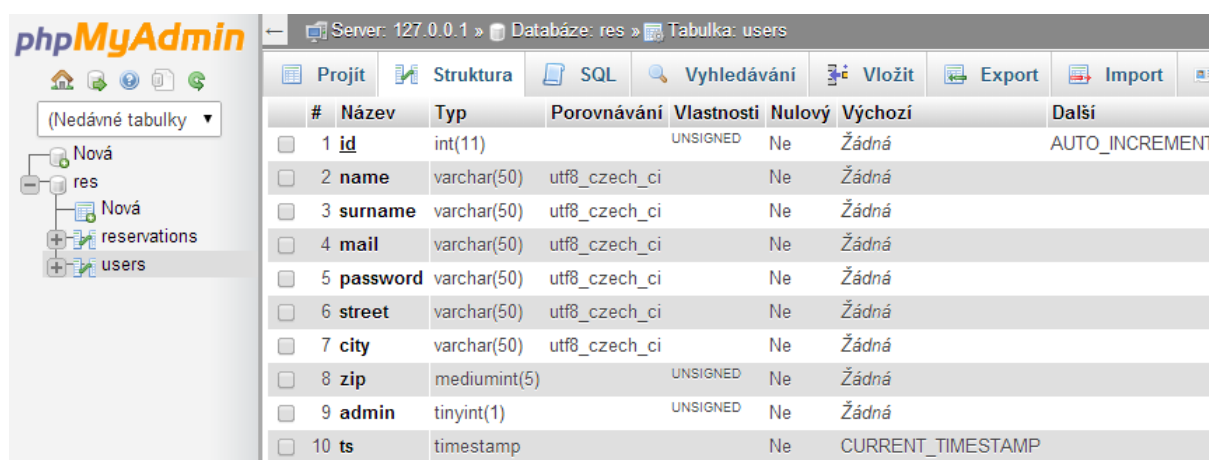
<?php
    generate(date("Y-m-d"));
?>
        </table>
    </div>
</div>
<div class="col-md-4 col-sm-6 col-lg-4"><h2>
<?php
    echo date("d.m.Y", strtotime(date("d.m.Y").'+1 day'));
?></h2>
    <div class="table-responsive">
        <table class="table table-hover">
            <th>Čas
            </th>
            <th>Rezervace
            </th>
<?php
    generate(date("Y-m-d", strtotime(date("Y-m-d").'+1 day')));
?>
        </table>
    </div>
</div>
<div class="col-md-4 col-sm-6 col-lg-4"><h2>
<?php
    echo date("d.m.Y", strtotime(date("d.m.Y").'+2 day'));
?></h2>
    <div class="table-responsive">
        <table class="table table-hover">
            <th>Čas
            </th>
            <th>Rezervace
            </th>
<?php
    generate(date("Y-m-d", strtotime(date("Y-m-d").'+2 day')));
?>
        </table>
    </div>
</div>
<div class="col-md-12 col-sm-12 col-lg-12">
    <div class="well well-sm">Copyright 2014 by Městské služby Litomyšl
s.r.o.
    </div>
</div>
</div>
</body>
</html>

```

Z uvedeného zdrojového kódu je patrné použití deklarace odpovídající specifikaci HTML 5. V rámci hlavičky stránky jsou připojeny potřebné soubory kaskádových stylů a JavaScript soubory. Dále jsou zde vloženy PHP soubory, které budou popsány níže. V rámci navigace je pomocí jazyka PHP rozhodováno, zdali je uživatel přihlášen a pokud ano, zdali se jedná o správce systému. Podle role uživatele je následně vložen soubor, obsahující HTML kód specifického menu. Dále je zde použit Bootstrap komponenta „carousel“, pomocí které je možné zobrazit navrženou fotogalerii. Dále

jsou v rámci tohoto dokumentu definovány tabulky, jejichž obsah je generován pomocí volání funkce „generate“. Tato funkce je definována v rámci vloženého souboru „tablegen.php“. Pomocí druhého vloženého PHP souboru „database.php“ je specifikováno připojení k databázi. Další HTML soubory jsou charakteristické strukturou, která je velmi podobná té uvedené, přičemž jsou využívány různé PHP soubory a jejich funkce.

Po dokončení základní struktury budoucího systému v podobě HTML dokumentů byla pomocí databázového systému MySQL vytvořena potřebná relační databáze. Tato databáze opět plně koresponduje s daným návrhem, který je reprezentován uvedeným relačním datovým modelem. Jsou zde tedy specifikovány dvě databázové tabulky. V rámci jedné tabulky budou uloženy informace o uživateli systému a v rámci druhé tabulky budou uloženy informace o jejich rezervacích. Dále zde byly u každé tabulky definovány její atributy a těm přiřazeny konkrétní datové typy a vlastnosti dle předchozího návrhu. Definice celé databáze, včetně obou tabulek a jejich atributů probíhala prostřednictvím nástroje phpMyAdmin. Na obrázku 4.5 je možné vidět návrhovou strukturu tabulky „users“ v prostředí phpMyAdmin.



#	Název	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Další
<input type="checkbox"/> 1	id	int(11)		UNSIGNED	Ne	Žádná	AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(50)	utf8_czech_ci		Ne	Žádná	
<input type="checkbox"/> 3	surname	varchar(50)	utf8_czech_ci		Ne	Žádná	
<input type="checkbox"/> 4	mail	varchar(50)	utf8_czech_ci		Ne	Žádná	
<input type="checkbox"/> 5	password	varchar(50)	utf8_czech_ci		Ne	Žádná	
<input type="checkbox"/> 6	street	varchar(50)	utf8_czech_ci		Ne	Žádná	
<input type="checkbox"/> 7	city	varchar(50)	utf8_czech_ci		Ne	Žádná	
<input type="checkbox"/> 8	zip	mediumint(5)		UNSIGNED	Ne	Žádná	
<input type="checkbox"/> 9	admin	tinyint(1)		UNSIGNED	Ne	Žádná	
<input type="checkbox"/> 10	ts	timestamp			Ne	CURRENT_TIMESTAMP	

Obr. 4.5 Struktura tabulky

V okamžiku vytvoření potřebné relační databáze bylo možné přikročit k definování základní funkcionality celého rezervačního systému. Nosnou komponentou jsou v tomto případě PHP skripty. Pomocí těchto skriptů bude zabezpečeno např. připojení k vytvořené databázi, generování tabulek rezervací, vytváření nových uživatelů, přihlašování uživatelů, vytváření uživatelských rezervací apod. V podstatě veškeré uživatelské požadavky budou obsluhovány na straně serveru, tudíž prostřednictvím PHP skriptů. Za tímto účelem bylo vytvořeno celkem 15

PHP souborů, které obsahují deklaraci jednotlivých skriptů. Do výše uvedeného zdrojového kódu jsou vloženy dva z těchto externích PHP souborů, jejichž obsah je zde uveden.

database.php

```
<?php
$db_connect = mysqli_connect('localhost','root','','res');
if (!$db_connect)
{
    header('Location: ../fail.php');
    exit();
}
?>
```

V rámci tohoto skriptu je zřízeno připojení k vytvořené databázi. Toto připojení je zabezpečeno prostřednictvím volání funkce „mysqli_connect“. Parametry této funkce jsou zde: adresa databázového serveru, přihlašovací jméno, heslo a název databáze. Vzhledem k tomu, že rezervační systém je tvořen v lokálním testovacím prostředí (prostřednictvím lokální instalace Apache serveru, MySQL a PHP), je jako adresa serveru uvedena hodnota „localhost“ a heslo není uvedeno. Pokud se připojení k databázi z nějakého důvodu nezdaří, je uživatel přesměrován na stránku „fail.php“.

tablegen.php

```
<?php
function generate($date)
{
    mysqli_query($GLOBALS['db_connect'], "SET NAMES 'utf8'");
    $result_object = mysqli_query($GLOBALS['db_connect'], "SELECT time,
id_user FROM reservations WHERE date = '$date'");
    $array_time = [];
    $array_id = [];
    $array_comb = [];
    $array_index = [];
    while ($row = mysqli_fetch_array($result_object))
    {
        $array_time[] = $row['time'];
        $array_id[] = $row['id_user'];
    }

    if (isset($_SESSION["id"]))
    {
        $array_comb = array_combine($array_time, $array_id);
        $array_index = array_keys($array_comb, $_SESSION["id"]);
        for ($int = 0; $int < 900; $int = $int + 90)
        {
            $firsttime = "07:00:00";
            $period = $int;
            $time = date('H:i:s', strtotime($firsttime . '+' . $period
. 'minutes'));
            if (in_array($time, $array_index) == true)
            {
```

```

        echo " <tr class=\"info\"><td>", date('H:i',
strtotime($firsttime . '+' . $period . 'minutes')), "</td><td>Vaše
rezervace</td></tr>";
    }
    else if (in_array($time, $array_time) == true)
    {
        echo " <tr class=\"danger\"><td>", date('H:i',
strtotime($firsttime . '+' . $period . 'minutes')),
"</td><td>Rezervováno</td></tr>";
    }
    else
    {
        echo " <tr><td>", date('H:i', strtotime($firsttime . '+' .
$period . 'minutes')) , "</td><td><form class=\"form-inline\" role=\"form\"
action=\"php/reservation.php\" method=\"post\"><input type=\"hidden\"
name=\"time\" value=\"\$time\"><input type=\"hidden\" name=\"date\"
value=\"\$date\"><button type=\"submit\" class=\"btn btn-xs btn-
success\">Rezervovat</button></form></td></tr>";
    }
}
}
else
{
    for ($int = 0; $int < 900; $int = $int + 90)
    {
        $firsttime = "07:00:00";
        $period = $int;
        $time = date('H:i:s', strtotime($firsttime . '+' . $period
. 'minutes'));
        if (in_array($time, $array_time) == true)
        {
            echo " <tr class=\"danger\"><td>", date('H:i',
strtotime($firsttime . '+' . $period . 'minutes')),
"</td><td>Rezervováno</td></tr>";
        }
        else
        {
            echo " <tr><td>", date('H:i', strtotime($firsttime . '+' .
$period . 'minutes')) , "</td><td>Volný termín</td></tr>";
        }
    }
}
}
?>

```

V uvedeném skriptu je definována funkce „generate“, jejímž parametrem je konkrétní datum, pro které se má tabulka termínů generovat. Nejprve je v rámci této funkce nastavena komunikace s databází na základě znakové sady UTF-8. Poté je pomocí SQL příkazu „SELECT“ vybrán čas rezervace a id uživatele, který rezervaci vytvořil. To vše je vybíráno z tabulky „reservations“, kde je na sloupci „date“ uloženo požadované datum. Poté jsou zde deklarována prázdná pole, která budou dále použita. Prostřednictvím cyklu „while“ a funkce „mysqli_fetch_array“ jsou naplněna dvě z těchto polí hodnotami získanými z databáze. V dalším kroku pokračuje provádění skriptu ve dvou větvích. Pokud je uživatel přihlášen (existuje superglobální proměnná

„SESSION[\"id\"]“), jsou naplněna zbývající dvě definovaná pole. Pomocí funkce „array_combine“ je naplněno první z nich, přičemž indexem je vždy čas a hodnotou je id uživatele. Poté je možné pomocí funkce „array_keys“ naplnit poslední definované pole hodnotami indexů, jejichž hodnoty jsou shodné s id přihlášeného uživatele. Tím je získáno potřebné pole, které obsahuje všechny rezervované časy konkrétního uživatele v rámci daného data. Poté jsou pomocí cyklu „for“ tisknuty jednotlivé řádky tabulky. Zde je potřebné nastavit počáteční časovou hodnotu (v případě zimního stadionu 7:00) a velikost jedné časové jednotky (90 minut). Časová jednotka je pak při každém průchodu cyklem zvětšena o hodnotu 90. Cyklus bude probíhat dokud, bude daná hodnota menší než 900. V praxi to znamená, že bude vytisknuto 10 řádků. Výhodou tohoto přístupu je to, že když bude potřeba změnit otevírací dobu stadionu, nebo délku jedné časové jednotky stačí přepsat uvedené hodnoty pouze na tomto místě. Jednotlivé tisknuté řádky pak obsahují dva sloupce. V prvním sloupci je zobrazen časový údaj (počáteční časová hodnota + časová jednotka) a ve druhém informace o obsazenosti daného termínu. Do druhého sloupce je možné vytisknout řetězce „Vaše rezervace“ a „Rezervováno“, nebo formulář se skrytými poli (obsahují časovou hodnotu na daném řádku a konkrétní datum). Pomocí tohoto formuláře je pak možné daný termín rezervovat. Samotná rezervace je však definována v rámci jiného PHP skriptu. To co bude vytisknuto na konkrétním řádku, závisí na tom, zdali se jedná o rezervaci přihlášeného uživatele, rezervaci jiného uživatele, nebo volný termín. Tyto údaje jsou zjišťovány pomocí ověřování výskytu daných hodnot ve dříve definovaných polích. Pokud uživatel není přihlášen, je možné vytisknout do druhého sloupce pouze řetězce „Volný termín“ a „rezervováno“.

Definováním PHP skriptů byla již vytvořena podstatná část celého systému. Bylo však zapotřebí zavést některé funkcionality na straně klienta. Za tímto účelem byla použita především knihovna jQuery. Nejzásadnější aplikací skriptování na straně klienta bylo zavedení kalendáře pro volbu požadovaného data. Tato funkcionality je poskytována také v rámci HTML 5 kde je možné v konkrétním atributu formuláře využít nového vstupního typu „date“. Vzhledem k tomu, že tento potřebný vstupní typ není zatím podporován ani v nejnovějších verzích některých internetových prohlížečů (např. Mozilla Firefox, Internet Explorer), bylo nutné zajistit tuto funkcionality pomocí jazyka JavaScript. Jako nejefektivnější se nakonec ukázalo použití hotového řešení. V tomto případě byla do vytvářeného systému integrována komponenta s názvem bootstrap-

datepicker.[bootstrap-datepicker] Tato komponenta využívá již zmíněných knihoven jQuery a Bootstrap a je šířena pod MIT licencí. Pomocí této komponenty je umožněno uživateli zvolit datum, ze zobrazeného kalendáře, což je jistě komfortnější než kdyby musel zadávat ručně požadované datum ve stanoveném formátu. Připojení potřebných JavaScript souborů probíhá standardně v hlavičce stránky.

```
<!DOCTYPE html>
<html lang="cs">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Rezervační systém zimního stadionu Litomyšl">
    <title>Zimní stadion Litomyšl
    </title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/bootstrap-theme.min.css" rel="stylesheet">
    <link href="css/datepicker.css" rel="stylesheet">
    <link href="css/styles.css" rel="stylesheet">
    <script src="js/jquery.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/bootstrap-datepicker.js"></script>
    <script src="js/bootstrap-datepicker.cs.js"></script>
  </head>
```

Zde jsou oproti předchozímu příkladu připojeny navíc soubory „datepicker.css“, „bootstrap-datepicker.js“ a „bootstrap-datepicker.cs.js“. Tyto soubory slouží postupně k definování zobrazení, funkcionality a české lokalizace uvedené komponenty. Samotné použití je pak umožněno prostřednictvím definování běžného formuláře, s jedním vstupním polem typu text a následného volání příslušné funkce v rámci elementu script. Této funkci je pak možné přiřadit několik parametrů. V tomto případě je důležitým parametrem „format“, pomocí kterého je možné zadat formát zobrazovaného data. Dále také „startDate“ a „endDate“, což jsou parametry, kterými je ohraničen rozsah vstupních dat (zde je do parametru „startDate“ přiřazeno aktuální datum). Parametrem „language“ je dále stanovena lokalizace daného kalendáře. Níže je uvedena ukázka zmiňovaného kódu.

```
<form class="form-inline" role="form" action="terms.php" method="post">
  <div class="form-group">
    <input type="text" class="datepicker" placeholder="Vyberte si datum" name="date">
  </div>
  <button type="submit" class="btn btn-info">Vybrat</button>
</form>
<script>
  $(''.datepicker').datepicker({
    format: "dd.mm.yyyy",
    startDate: "<?php echo $startdate;?>",
```

```
        endDate: "01/03/2015",  
        language: "cs",  
        todayBtn: true,  
        todayHighlight: true,  
        autoclose: true, });  
</script>
```

V dalším průběhu implementace nového rezervačního systému byly použity další dvě podobné komponenty. Jednou z nich je `jQueryValidation`, což je opět komponenta využívající knihovnu `jQuery` a je šířena pod MIT licenci.[`jQueryValidation`] Jak již její název napovídá, je pomocí ní možné ověřovat validitu polí vstupních formulářů. V tomto projektu je toto využito pro ověření dat v registračním formuláři. Dalším využitým modulem je `tablesorter`.[`tablesorter`] Tento modul je opět šířen pod stejnou licenci a využívá `jQuery` knihovnu. Je možné ho využít pro vzestupné a sestupné řazení údajů v jednotlivých sloupcích konkrétních tabulek. V projektu rezervačního systému je využit například k řazení uživatelských rezervací podle příjmení uživatele, nebo podle data rezervace. Použití obou zmíněných komponent je velmi podobné jako v uvedeném příkladu. Tedy jsou v hlavičce dokumentu připojeny příslušné JavaScript soubory a na požadovaném místě jsou volány konkrétní funkce s danými parametry.

4.2.1. Integrace termínů rezervací do kalendářů různých platforem

Poté co byly implementovány veškeré zmiňované komponenty, lze považovat rezervační systém hrací plochy za vytvořený. V tomto okamžiku byly splněny veškeré uživatelské požadavky na funkcionalitu i vzhled daného systému. Vzhledem k tomu, že se má, ale jednat o moderní rezervační systém, byla na závěr implementační části naprogramována ještě dodatečná funkce. Pomocí této funkce má být zabezpečena integrace vytvořeného systému s moderními kalendářovými aplikacemi. Hlavním motivem tohoto kroku je uživatelská potřeba synchronizovaného kalendáře, přičemž tato potřeba nadále stoupá s rozvojem operačních systému pro mobilní zařízení. V dnešní době je naprosto běžné spravovat veškeré události v rámci jednoho kalendáře. To v praxi znamená, že je uživateli umožněno spravovat konkrétní události vytvořené například v mobilní aplikaci prostřednictvím webového prohlížeče, nebo kalendářní aplikace konkurenční platformy. Z tohoto důvodu by bylo dobré, kdyby bylo uživateli vytvořeného rezervačního systému umožněno, svou rezervaci integrovat do libovolného kalendáře. Toho bylo nakonec také dosaženo a to prostřednictvím standardu `iCalendar`.

iCalendar je zkrácený název normy RFC 2445 - Internet Calendaring and Scheduling Core Object Specification, jejíž nejnovější verzi vydala v roce 2009 komise IETF. Jedná se o specifikaci pro výměnu kalendářových mezi aplikacemi a systémy. Definice kalendáře je podle tohoto standardu zapisována prostým textem v rámci souboru s příponou „.ics“. Formát zápisu je pak poměrně striktní, přičemž všechna data týkající se daného kalendáře jsou umístěna v tzv. objektu kalendáře. Samotný objekt kalendáře může obsahovat několik definovaných komponent:

- To-do – komponenta pro definování úkolu,
- Event – komponenta pro definování události,
- Journal – komponenta pro přidávání zápisu do deníku,
- Free/Busy – komponenta pro zjištění obsazenosti v konkrétním čase,
- TimeZone – komponenta pro popis časové zóny kalendáře,
- Alarm – komponenta pro definování připomenutí.

Co se týče samotného zápisu objektu kalendáře, specifikace zde určuje následující tvar:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
END:VCALENDAR
```

Položky „VERSION“ a „PRODID“ jsou povinné položky a postupně udávají verzi kalendáře a globálně unikátní identifikátor, který specifikuje tvůrce kalendáře. Dále je povinné uvést specifikaci alespoň jedné kalendářové komponenty. V případě uvedení komponenty Event je pak zápis celého objektu kalendáře podle uvedené normy následující:

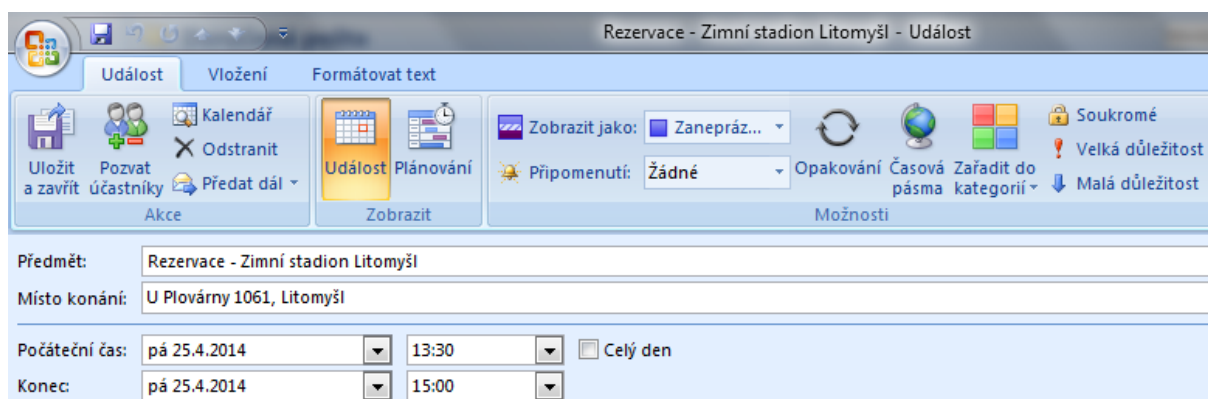
```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
BEGIN:VEVENT
UID:19970610T172345Z-AF23B2@example.com
DTSTAMP:19970610T172345Z
DTSTART:19970714T170000Z
DTEND:19970715T040000Z
SUMMARY:Bastille Day Party
END:VEVENT
END:VCALENDAR
```

Položka „BEGIN“ udává, že se jedná o komponentu Event, v položce „UID“ je definován jednoznačný identifikátor události a v následujících položkách je pak specifikován přesný čas vytvoření, začátku a konce dané události a její popis. Formát zápisu času je v tomto standardu jasně specifikován a je nutné ho dodržet. Písmeno Z na konci zápisu udává, že se jedná o čas UTC (Coordinated Universal Time). Zápis může obsahovat ještě další položky, jako například „LOCATION“, nebo „GEO“. [IETF – iCalendar]

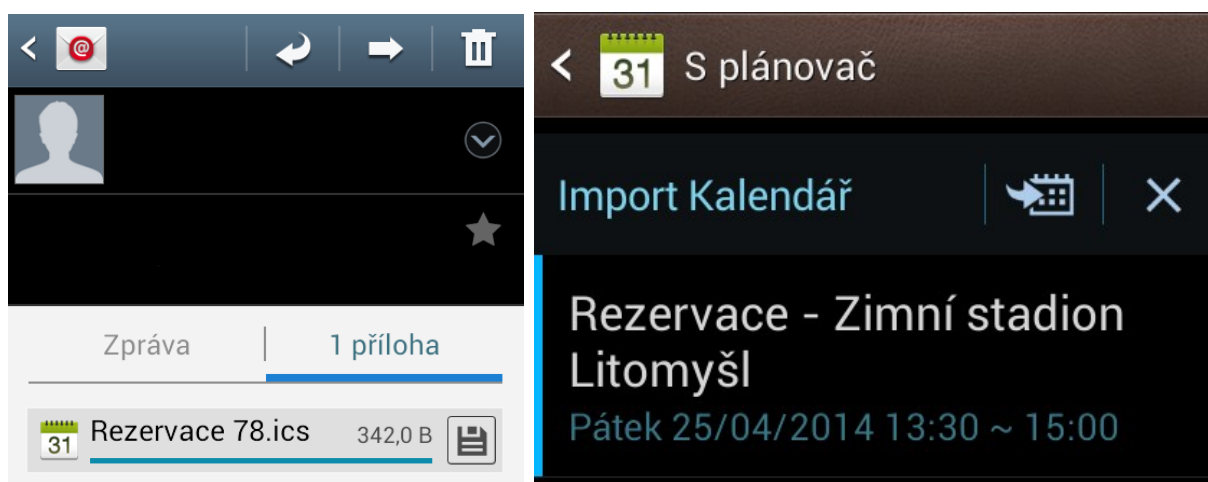
Vzhledem k tomu, že je uvedený standard podporován většinou hlavních kalendářových aplikací, jeví se jeho použití v případě integrace rezervačního systému jako správné rozhodnutí. Specifikace iCalendar je podporována například aplikacemi: Apple Calendar, Google Calendar, Microsoft Office Outlook, Windows Calendar, Facebook a spoustou dalších. Pro sestavení souboru nesoucího objekt kalendáře byl v rámci vytvořeného rezervačního systému definován PHP skript, který dále daný soubor odešle uživateli na zadanou e-mailovou adresu. Pro odeslání souboru byla použita volně dostupná komponenta Swift Mailer.[swiftmailer] Po přijetí iCalendar souboru je možné integrovat definovanou událost do libovolného kalendáře podporujícího tento standard. Integrace je ve většině případů možná přímo z prostředí e-mailového klienta, tak je zachyceno na obrázcích 4.6 a 4.7. Struktura daného souboru, který je rezervačním systémem generován, je pak následující:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//mslit.cz/ZS//NONSGML v1.0//cs-CZ
BEGIN:VEVENT
UID:20140424T114608Z@zimniStadionLitomysl
DTSTAMP:20140424T114608Z
DTSTART:20140425T113000Z
DTEND:20140425T130000Z
SUMMARY:Rezervace - Zimní stadion Litomyšl
LOCATION:U Plovárny 1061, Litomyšl
GEO:49.86812;16.319042
END:VEVENT
END:VCALENDAR
```

Položky „UID“, „DTSTAMP“, „DTSTART“ a „DTEND“ jsou generovány na straně serveru, na základě údajů, o konkrétní uživatelské rezervaci.



Obr. 4.6 Zpracování souboru iCalendar v prostředí Microsoft Outlook 2007



Obr. 4.7 Zpracování iCalendar – mobilní zařízení, Obr. 4.8 Import do kalendáře - mobilní zařízení

Jak je patrné z uvedených obrázků (Obr. 4.6, 4.7, 4.8), zpracování souboru iCalendar je možné na rozdílných platformách i v různých aplikacích. Na obrázku 4.6 je zobrazeno zpracování přijatého souboru v prostředí Microsoft Outlook 2007 (platforma Microsoft Windows 7), kde je možné ihned událost importovat do příslušného kalendáře. Z tohoto vyplývá, že podpora uvedeného standardu je implementována i ve starších verzích e-mailových a kalendářových aplikací. Na dalších obrazech (4.7 a 4.8) je zobrazeno zpracování přijatého souboru v prostředí defaultního e-mailového klienta operačního systému Android (verze 4.1) mobilního zařízení značky Samsung a jeho následný možný import do aplikace Google Calendar.

Poté co byla zabezpečena možnost integrace uživatelských rezervací do nejpoužívanějších kalendářových aplikací, lze považovat implementaci rezervačního systému hrací plochy zimního stadionu za dokončenou.

5. Zhodnocení navrhovaného řešení

V předchozích kapitolách byly provedeny potřebné analýzy a vytvořeny konkrétní návrhy budoucího rezervačního systému hrací plochy zimního stadionu. Poté bylo teprve přikročeno k jeho samotné implementaci. Tímto postupem bylo zajištěno vytvoření kvalitního systému, který bude přínosem jak pro samotnou společnost spravující zimní stadion, tak pro její klienty a případné zájemce o rezervaci hrací plochy. Implementace celého systému byla provedena za použití moderních technologií a postupů, uváděných v odborné literatuře. V této kapitole bude již vytvořený systém posouzen z hlediska jeho konkrétních přínosů a ekonomických souvislostí. Tyto přínosy budou uváděny nejprve z pohledu společnosti spravující zimní stadion a poté z pohledu samotného uživatele systému.

Z pohledu správcovské společnosti je hlavním přínosem navrhovaného systému, automatizace celého procesu rezervace hrací plochy. V současnosti používaný systém rezervace postrádá jakýkoliv prvek automatizace a je tak pro společnost značně neefektivní. Připomeňme, že současný proces rezervace probíhá zcela manuálně. Správce zimního stadionu, nebo jím pověřený zaměstnanec, je nucen být po celou pracovní dobu přítomen v kanceláři zimního stadionu, aby mohl vyřizovat telefonické, nebo osobní požadavky na rezervace daných termínů.

Zavedením navrhovaného elektronického rezervačního systému tato nutnost odpadá a je zapotřebí pouze minimální administrativa ze strany správce zimního stadionu. To by správcovské společnosti přineslo značnou úsporu nákladů v podobě mezd daných zaměstnanců, nebo jejich možného využití při jiné činnosti. Tento přínos je tak snadno vyčíslitelnou finanční úsporou. Dalším vyčíslitelným přínosem je pak zvýšení konkurenceschopnosti a atraktivity zimního stadionu v případě zavedení navrhovaného řešení. Vezmeme-li v potaz fakt, že v okolí zimního stadionu (okruh 30km) se nachází několik podobných sportovních zařízení, je velmi důležité nabídnout případným klientům nějakou přidanou hodnotu. Tou může bezesporu být elektronický rezervační systém hrací plochy. Příliv nových zájemců o rezervaci hrací plochy se pak prokazatelně odrazí na zvýšení finančního příjmu správcovské společnosti. Zavedením navrhovaného systému je tedy možné uspořít náklady na straně jedné a zvýšit příjmy na straně druhé. Z dlouhodobého hlediska je možné za přínos označit také zlepšení celkového obrazu správcovské společnosti v očích veřejnosti i jejích

partnerů. Z tohoto pohledu se jedná především o udržení dobrého jména prostřednictvím zavádění moderních technologií a služeb pro své klienty.

Z pohledu zájemce o rezervaci hrací plochy zimního stadionu, jakožto uživatele rezervačního systému, je pak hlavním přínosem navrhovaného řešení značné zjednodušení celého procesu rezervace. V současnosti je klient nucen telefonicky, nebo osobně kontaktovat správce zimního stadionu, nebo jím pověřenou osobu, s požadavkem na rezervaci hrací plochy. Tento postup může být značně nekomfortní a to hlavně z toho důvodu, že zájemce nemá k dispozici náhled do aktuální tabulky obsazenosti jednotlivých termínů. Tato tabulka je sice umístěna k náhledu v prostorách zimního stadionu, avšak v případě telefonického hovoru je zájemce zcela odkázán na informace poskytnuté ústní formou. Tento fakt může zapříčinit opakované prodlužování telefonické komunikace v případě, že je uživatel nucen se několikrát dotazovat na obsazenost požadovaných termínů. Pro osobní kontakt zájemce se správcem je pak nutné vážit cestu na zimní stadion. Obě nyní dostupné varianty jsou z pohledu zájemce do značné míry obtěžující.

V případě zavedení navrhovaného řešení je zájemci umožněn další způsob rezervace. Elektronický rezervační systém je pro uživatele velmi pohodlné řešení, u kterého odpadají předchozí nevýhody. Tedy uživatel má kdykoliv a odkudkoliv k dispozici velmi detailní přehled obsazenosti jednotlivých termínů a to bez jakýchkoliv dodatečných finančních nákladů. Požadované termíny je pak možné velmi rychle a snadno rezervovat. Důležitým přínosem je také možnost správy vytvořených rezervací. To v praxi znamená, že má uživatel možnost kdykoliv prohlížet a případně rušit vytvořené rezervace. K celkové spokojenosti uživatele s navrhovaným systémem by měla přispět také vysoká intuitivnost jeho ovládání.

V předešlých odstavcích byly zhodnoceny hlavní přínosy navrhovaného řešení z pohledu zadavatelské společnosti a uživatele systému. Samozřejmostí je, že na zavedení tohoto řešení musí být vynaloženy také určité zdroje. Z pohledu správcovské společnosti se tyto zdroje týkají především finančních nákladů na samotné zavedení a následný provoz systému. Vzhledem k použití navrhovaných technologií, by neměl být problém systém provozovat na stejném serveru, na kterém jsou nyní nasazeny firemní webové stránky (podpora PHP, MySQL). Samotné nasazení systému by si tedy mělo vyžádat pouze náklady spojené s prací specialisty, který by daný systém zaváděl a náklady spojené se školením personálu. Co se týče provozu systému, je situace jiná.

V případě neexistence dodavatelské podpory by byly náklady na provoz systému plně hrazeny danou společností. Přičemž je třeba počítat s faktem, že postupem času se budou náklady na provoz systému zvyšovat, protože bude nutné přizpůsobovat ho modernějším technologiím.

Vhodnější alternativou pro správcovskou společnost by mohl být provoz systému jako služby. V tomto případě by byl celý systém nasazen na serveru dodavatele a správcovská společnost by platila určený poplatek za jeho užívání. Tím by odpadly jakékoliv náklady spojené se zavedením a údržbou systému ze strany dané společnosti. Veškeré náklady spojené s rezervačním systémem jsou pak reprezentovány nejčastěji měsíčním poplatkem za provoz systému. Vzhledem k tomu, že podobná řešení jsou běžně nabízena v řádu stokorun měsíčně, jeví se tato možnost jako nejvhodnější alternativa. Při rozhodování o způsobu nasazení nového řešení musejí být v rámci dané společnosti konkrétně vyčísleny uvedené možné příjmy a úspory a tyto pak porovnány s možnými uvedenými náklady na tento systém.

6. Závěr

Na konci této diplomové práce je implementován kompletní elektronický rezervační systém hrací plochy zimního stadionu. Tohoto bylo dosaženo prostřednictvím praktické aplikace teoretických poznatků z oblasti internetových technologií a informačních systémů. Celá práce byla rozdělena do několika stěžejních kapitol, ve kterých je tento postup zachycen. V této závěrečné kapitole bude stručně zrekapitulován obsah jednotlivých předchozích kapitol a uveden jejich význam v rámci této diplomové práce.

V úvodní kapitole byly vymezeny základní pojmy z oblasti informačních technologií, zejména pak z prostředí internetu a jeho služeb. Byla zde stručně shrnuta historie internetu a jeho klíčové služby World Wide Web. Dále zde byly zavedeny pojmy HTTP a HTML, jakožto základní technologie pro fungování této služby. Poté byly uvedeny rozdíly mezi statickými a dynamickými internetovými stránkami, přičemž byla řeč o serverových skriptovacích jazycích a klientských serverových jazycích. Dále byl charakterizován význam nasazení informačních technologií v současných firmách a na závěr úvodní kapitoly byl stanoven hlavní cíl této diplomové práce.

Následující kapitola byla věnována především teoretickým přístupům a nástrojům týkajících se dané problematiky. Byl zde charakterizován značkovací jazyk HTML, jakožto hlavní technologie pro tvorbu internetových stránek. Byla zde uvedena historie tohoto jazyka a také postup jeho standardizace v prostředí organizace W3C. Dále zde byly uvedeny verze HTML a také byl specifikován jazyk XHTML. Zmíněna byla také nejnovější specifikace HTML 5 a její nejvýznamnější vlastnosti. Stejně jako v případě HTML, byla uvedena také stručná historie kaskádových stylů a stanoven jejich význam a použití. Dále byly uvedeny ukázky zápisu CSS a prostor byl věnován také nejnovější specifikaci CSS 3 a jejím novým vlastnostem. Dále byl v této kapitole charakterizován klientský skriptovací jazyk JavaScript, je uvedena jeho historie a specifikovány tři jeho součásti (ECMAScript, DOM, BOM). Dále je v této kapitole zmíněna také knihovna jQuery, založená právě na jazyku JavaScript. Jako zástupce serverových skriptovacích jazyků je pak uveden jazyk PHP. Podkapitola věnovaná této technologii se týká především specifikace fungování PHP a procesu zpracovávání požadavků na straně serveru. Poslední uváděnou technologií je zde MySQL. V této souvislosti jsou zavedeny pojmy jako: systém řízení báze dat, relační model a

databázový systém. Prostor je věnován také specifikaci jazyka SQL, jakožto nástroje pro práci nad daty relačních databází. Uvedeny zde byly také základní příkazy tohoto jazyka. Závěr této kapitoly je věnován teorii týkající se projektování informačních systémů. Nejprve je vymezen pojem informační systém a poté jsou stanoveny etapy životního cyklu jeho vývoje. Dále jsou tyto etapy podrobně charakterizovány. Cílem celé této kapitoly bylo položení teoretického základu pro další kapitoly této diplomové práce. Byly zde specifikovány obecné postupy a nástroje, které byly dále použity při samotném vytváření rezervačního systému.

V pořadí třetí kapitola byla zaměřena na provedení analýzy současného stavu a specifikaci požadavků zadavatele. V úvodu této kapitoly byl stručně charakterizován objekt zadavatele, jímž je společnost Městské služby Litomyšl s.r.o. Tato společnost spravuje zimní stadion v Litomyšli, pro který je rezervační systém vytvářen. Poté bylo přikročeno k samotné analýze současného stavu, kde byl charakterizován stávající systém rezervace hrací plochy zimního stadionu. Tento systém byl z několika důvodů shledán nevyhovujícím. Následně proběhla specifikace požadavků zadavatele. Tyto požadavky byly sestaveny na základě rozhovorů s jednatelem správcovské společnosti a správcem zimního stadionu. Výsledné požadavky byly poté rozděleny do dvou skupin a to jako požadavky na vzhled systému a požadavky na funkce systému. Na základě daných požadavků byla následně sestavena tabulka uživatelských požadavků. Pro jednotlivé uživatelské požadavky byly sestaveny také případy užití, díky kterým bylo možné získat lepší pohled na rozsah celého systému. Pro přehlednost byl v závěru této kapitoly uveden sestavený diagram aktivit. Cílem třetí kapitoly bylo správně analyzovat stávající systém a sestavit uživatelské požadavky. Veškeré uvedené techniky pak měly napomoci ke správnému pochopení fungování vytvářeného systému. Na těchto základech pak bylo možné sestavit konkrétní návrhy daného systému.

Další kapitola se týkala návrhů a samotné implementace řešeného rezervačního systému. V rámci návrhů byl nejprve sestaven relační datový model, pomocí kterého byla stanovena možná struktura databázových tabulek relační databáze. Dále byly sestaveny grafické návrhy a to především návrh rozložení webové stránky a návrh vzhledu systému. Sestaveny byly také návrhy vstupních formulářů a výstupních sestav. Po odsouhlasení vytvořených návrhů zadavatelem projektu, bylo přistoupeno ke specifikaci systému z pohledu použitých technologií. Dále byl prostor

věnován implementační části. Zde byl již vytvářen konkrétní zdrojový kód pro jednotlivé komponenty systému. Nejprve byly vytvořeny jednotlivé HTML dokumenty, jako prezentační část celého systému. Finální zobrazovaný vzhled jednotlivých dokumentů byl definován prostřednictvím CSS. V této fázi byl použit nástroj Bootstrap a jeho komponenty. V následujícím kroku byla vytvořena konkrétní MySQL databáze a to za použití nástroje phpMyAdmin. Poté již bylo přikročeno k definování stěžejní funkcionality celého systému a to prostřednictvím skriptů psaných v jazyku PHP. Pro dosažení všech potřebných funkcí pak byly použity také komponenty využívající jazyku JavaScript a knihovny jQuery. V závěru této kapitoly byl představen standard iCalendar, jakožto prostředek vhodný pro integraci vytvářených uživatelských rezervací do moderních kalendářových aplikací. Zajištěním této integrace byl překročen rámec požadované funkcionality vytvořeného systému a byly tak vytvořena přidaná hodnota jeho uživatelům

V poslední kapitole byl vytvořený rezervační systém zhodnocen a to především z pohledu jeho přínosů. Ty byly rozděleny na přínosy pro uživatele systému a přínosy pro zadavatelskou společnost. Byly zde uvedeny také ekonomické souvislosti v podobě možného zvýšení příjmů i úspor pro provozovatele navrhovaného systému. Nakonec byly nastíněny možnosti jeho nasazení a provozu a to především z pohledu finančních nákladů.

Vzhledem k tomu, že tato diplomová práce je zaměřena především na návrh a následnou implementaci rezervačního systému, nebylo zde prováděno dostatečné testování a následné nasazení systému. Testování probíhalo pouze na zkušebních datech v rámci zkušebního provozu. Bylo by tedy možné na tuto práci dále navázat a to především specifikací následných fází, určených životním cyklem vývoje informačního systému. Tyto fáze jsou například: komplexní testování systému, instalace systému, finalizace dokumentace systému, školení uživatelů a další fáze spadající do etapy údržby systému.

Hlavním cílem této diplomové práce bylo navrhnout a vytvořit elektronický rezervační systém hrací plochy zimního stadionu v Litomyšli. Tohoto cíle mělo být dosaženo za použití moderních nástrojů, postupů a technologií. V závěru této diplomové práce lze konstatovat, že daný cíl byl zcela naplněn.

Seznam použité literatury

Tištěné zdroje

HOGAN, Brian P. *HTML5 a CSS3: výukový kurz webového vývojáře*. Vyd. 1. Brno: Computer Press, 2011, 272 s. ISBN 978-80-251-3576-1.

CHAFFER, Jonathan. *Mistrovství v jQuery: kompletní průvodce vývojáře*. 1. vyd. Brno: Computer Press, 2013, 384 s. ISBN 978-80-251-4103-8.

KALUŽA, Jindřich. *Informační systémy pro strategické řízení: bez předchozích znalostí : [průvodce pro samouky]*. 1. vyd. Ostrava: Vysoká škola báňská - Technická univerzita, Fakulta ekonomická, c2010, 145 s. ISBN 978-80-248-2280-8.

KALUŽA, Jindřich a Ludmila KALUŽOVÁ. *Modelování dat v informačních systémech: průvodce webového programátora*. 1. vyd. Praha: Ekopress, 2012, 125 s. ISBN 978-80-86929-81-1.

KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2007, 607 s. ISBN 978-80-251-1813-9.

MARGORÍN, Marián. *JQuery bez předchozích znalostí*. Vyd. 1. Brno: Computer Press, 2011, 253 s. ISBN 978-80-251-3379-8.

NIXON, Robin. *Learning PHP, MySQL, JavaScript, and CSS*. 2nd ed. Sebastopol, CA: O'Reilly, 2012, xxi, 556 p. ISBN 14-493-1926-2.

PONKRÁC, Miloslav. *PHP a MySQL: bez předchozích znalostí : [průvodce pro samouky]*. Vyd. 1. Brno: Computer Press, 2007, 221 s. ISBN 978-80-251-1758-3.

SCHAFER, Steven M. *HTML, XHTML a CSS: bible [pro tvorbu WWW stránek] : 4. vydání*. 1. vyd. Praha: Grada, 2009, 647 s. ISBN 978-80-247-2850-6.

ZAKAS, Nicholas C. *JavaScript pro webové vývojáře: programujeme profesionálně*. Vyd. 1. Překlad Lukáš Krejčí. Brno: Computer Press, 2009, 832 s. ISBN 978-80-251-2509-0.

Elektronické zdroje

Bootstrap [online]. [cit. 2014-03-02]. Dostupné z: <http://getbootstrap.com/>

Bootstrap-datepicker [online]. [cit. 2014-03-10]. Dostupné z: <http://bootstrap-datepicker.readthedocs.org/en/release/#>

JqBootstrapValidation [online]. [cit. 2014-03-10]. Dostupné z: <http://reactiveraven.github.io/jqBootstrapValidation/>

Candidate Recommendation 04 February 2014 [online]. 2014 [cit. 2014-04-03]. Dostupné z: <http://www.w3.org/TR/html>

Cascading Style Sheets (CSS) Snapshot 2010 [online]. 2010 [cit. 2014-04-05].
Dostupné z: <http://www.w3.org/TR/CSS/>

CSS3 Introduction [online]. [cit. 2014-04-06]. Dostupné z:
http://www.w3schools.com/css/css3_intro.asp

Document Object Model (DOM) Technical Reports [online]. [cit. 2014-04-08].
Dostupné z: <http://www.w3.org/DOM/DOMTR>

Encyclopædia Britannica [online]. [cit. 2014-04-12]. Dostupné z:
<http://www.britannica.com/EBchecked/topic/287895/information-system>

HTML5 New Elements [online]. [cit. 2014-04-04]. Dostupné z:
http://www.w3schools.com/html/html5_new_elements.asp

HTML5 Semantic Elements [online]. [cit. 2014-04-04]. Dostupné z:
http://www.w3schools.com/html/html5_semantic_elements.asp

ICalendar [online]. 2009 [cit. 2014-04-18]. Dostupné z:
<http://tools.ietf.org/html/rfc5545>

Introduction to SQL [online]. [cit. 2014-04-10]. Dostupné z:
http://www.w3schools.com/sql/sql_intro.asp

Swiftmailer [online]. [cit. 2014-04-15]. Dostupné z: <http://swiftmailer.org/>

Tablesorter [online]. [cit. 2014-04-08]. Dostupné z: <http://tablesorter.com/docs/>

XHTML2 Working Group Home Page [online]. 2007 [cit. 2014-04-05]. Dostupné z:
<http://www.w3.org/MarkUp/>

Seznam zkratek

AJAX – Asynchronous JavaScript and XML, technologie vývoje interaktivních webových aplikací

BOM – Browser Object Model, rozšíření JavaScriptu specifické pro webový prohlížeč

CASE – Computer-aided Software Engineering, použití softwaru při vývoji počítačových programů

CDN – Content Delivery Network, distribuovaný systém serverů

CGI – Common Gateway Interface, protokol pro propojení aplikací s webovým serverem

CR – Candidate Recommendation, stupeň schvalování W3C

CSS – Cascading Style Sheets, kaskádové styly

DBMS – DataBase Management System, systém řízení báze dat

DOM – Document Object Model, objektový model dokumentu

DTD – Document Type Definition, jazyk pro popis dokumentu

ECMA – European Computer Manufacturers Association, standardizační organizace

FK – Foreign Key, cizí klíč

GML – Generalized Markup Language, univerzální značkovací jazyk

HTML – HyperText Markup Language, značkovací jazyk pro hypertext

HTTP – Hyper Text Transfer Protocol, protokol pro výměnu hypertextových dokumentů

iCalendar – Internet Calendaring and Scheduling Core Object Specification, standard pro výměnu kalendářových dat

IETF – Internet Engineering Task Force, standardizační komise

IP – Internet Protocol, internetový protokol

IS – Information System, informační systém

ISO – International Organization for Standardization, standardizační organizace

IT – Information Technology, informační technologie

MIME – Multipurpose Internet Mail Extensions, internetový standard

MIT – Massachusetts Institute of Technology, svobodná licence, která vznikla na Massachusettském technologickém institutu

MSL – Městské služby Litomyšl

PHP – PHP: Hypertext Preprocessor, serverový skriptovací jazyk

PK – Primary Key, primární klíč

PR – Proposed Recommendation, stupeň schvalování W3C

PV – Present Value, současná hodnota

REC – Recommendation, stupeň schvalování W3C

ROI – Return on Investment, návratnost investic

SGML – Standard Generalized Markup Language, univerzální značkovací jazyk

SQL – Structured Query Language, dotazovací jazyk používaný pro práci nad relační databází

URL – Uniform Resource Locator, specifikace umístění zdroje v rámci internetu

UTC – Coordinated Universal Time, koordinovaný světový čas

W3C – World Wide Web Consortium, mezinárodní konsorcium pro vývoj webových standardů

WD – Working Draft, stupeň schvalování W3C

WWW – World Wide Web

XHTML – eXtensible Hypertext Markup Language, značkovací jazyk pro hypertext

XML – Extensible Markup Language, obecný značkovací jazyk

Prohlášení o využití výsledků diplomové práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 - školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že bibliografické údaje o diplomové práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 23. 4. 2014



Bc. Aleš Kalousek

Seznam příloh

Příloha č.1: Vytvořené soubory na CD